



DDDDDDDD DDDDDDDDD DD DD LL DDDDDDDDD RRRRRRRR RRRRRRRR VV VV EEEEEEEEEE RRRRRRRR  
DDDDDDDD DDDDDDDDD DD DD LL DDDDDDDDD RRRRRRRR RRRRRRRR VV VV EEEEEE RRRRRRRR  
DD DD DD LL DD DD RR RR VV VV EE RR RR  
DD DD DD LL DD DD RR RR VV VV EE RR RR  
DD DD DD LL DD DD RR RR VV VV EE RR RR  
DD DD DD LL DD DD RRRRRRRR VV VV EEEEEE RRRRRRRR  
DD DD LL DD DD RRRRRRRR VV VV EEEEEE RRRRRRRR  
DD DD LL DD DD RR RR VV VV EE RR RR  
DD DD LL DD DD RR RR VV VV EE RR RR  
DD DD LL DD DD RR RR VV VV EE RR RR  
DD DD LL DD DD RR RR VV VV EE RR RR  
DD DD LL DDDDDDDDD RR RR VV VV EEEEEE RRRRRRRR  
DDDDDDDD DDDDDDDDD LL LL LLLLLLLL DDDDDDDDD RR RR VV VV EEEEEE RRRRRRRR

LL IIIII SSSSSSSS  
LL IIIII SSSSSSSS  
LL SS SS  
LL LLLLLLLL IIIII SSSSSSSS SSSSSSSS

(1)	113	EXTERNAL AND LOCAL DEFINITIONS
(1)	299	STANDARD TABLES
(1)	522	CONTROLLER INITIALIZATION ROUTINE
(1)	572	UNIT INITIALIZATION ROUTINE
(1)	695	DRIVER SPECIFIC SUBROUTINES
(1)	718	FDT ROUTINE - TEST TRANSFER BYTE COUNT ALIGNMENT
(1)	754	START I/O ROUTINE
(1)	1532	INTERRUPT SERVICE ROUTINE
(1)	1593	REGISTER DUMP ROUTINE
(1)	1632	MOVE TO USER BUFFER ROUTINE
(1)	1676	MOVE FROM USER BUFFER ROUTINE

0000 1 .TITLE DLDRIVER - VAX/VMS RL11/RL01,RL02 DISK DRIVER  
0000 2 .IDENT 'V04-000'  
0000 3 \*\*\*\*\*  
0000 4 \*\*\*\*\*  
0000 5 \*  
0000 6 \* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
0000 7 \* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
0000 8 \* ALL RIGHTS RESERVED.  
0000 9 \*  
0000 10 \* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
0000 11 \* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
0000 12 \* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
0000 13 \* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
0000 14 \* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
0000 15 \* TRANSFERRED.  
0000 16 \*  
0000 17 \* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
0000 18 \* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
0000 19 \* CORPORATION.  
0000 20 \*  
0000 21 \* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
0000 22 \* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
0000 23 \*  
0000 24 \*  
0000 25 \*\*\*\*\*  
0000 26 \*  
0000 27 \* FACILITY:  
0000 28 \*  
0000 29 \* VAX/VMS RL11/RL01,RL02 DISK DRIVER  
0000 30 \*  
0000 31 \* AUTHOR:  
0000 32 \*  
0000 33 \* C. FRANKS 05-OCT-1979  
0000 34 \*  
0000 35 \* MODIFIED BY:  
0000 36 \*  
0000 37 \*  
0000 38 \* V03-008 WHM0001 Bill Matthews 15-May-1984  
0000 39 \* Added MicroVAX I/QBUS support.  
0000 40 \*  
0000 41 \* V03-007 RAS0300 Ron Schaefer 27-Apr-1984  
0000 42 \* Add DEV\$M\_NNM characteristic to DECHAR2 so that these  
0000 43 \* devices will have the "node\$" prefix.  
0000 44 \*  
0000 45 \* V03-006 PRD0033 Paul R. DeStefano 09-Sep-1983  
0000 46 \* Added EXESLCLDSKVALID to function decision table.  
0000 47 \*  
0000 48 \* V03-005 ROW0211 Ralph O. Weber 16-AUG-1983  
0000 49 \* Change device-dependent UCB definition base from UCBSW\_BCR+2  
0000 50 \* to UCB\$K\_LCL\_DISK\_LENGTH.  
0000 51 \*  
0000 52 \* V03-004 KDM0059 Kathleen D. Morse 14-Jul-1983  
0000 53 \* Change time-wait loops to use new TIMEDWAIT macro.  
0000 54 \*  
0000 55 \* V03-003 PRD0020 Paul R. DeStefano 26-Apr-1983  
0000 56 \* Modified FATALERR routine to return SSS\_PARITY only for  
0000 57 \* errors that possibly indicate bad media. All other error

DLDRIEVER  
V04-000

- VAX/VMS RL11/RL01,RL02 DISK DRIVER<sup>I 5</sup> 16-SEP-1 14:00:17:29 VAX/VMS Macro V04-00  
5-SEP- : 00:12:24 [DRIVER.SRC]DLDRIEVER.MAR;1

Page 2  
(1)

0000	58	:	conditions which formerly returned SSS_PARITY now return
0000	59	:	SSS_CNTLERR.
0000	60	:	
0000	61	:	V03-002 KDM0002 Kathleen D. Morse 28-Jun-1982
0000	62	:	Added \$DYNDEF.
0000	63	:	
0000	64	:	V03-001 KTA0100 Kerbey T. Altmann 07-Jun-1982
0000	65	:	Add code to set UCB\$L_MEDIA_ID.
0000	66	:	
0000	67	**	

DL  
VC

0000 69 : ABSTRACT:  
0000 70  
0000 71 THIS MODULE CONTAINS THE TABLES AND ROUTINES NECESSARY TO  
0000 72 PERFORM ALL DEVICE-DEPENDENT PROCESSING OF AN I/O REQUEST  
0000 73 FOR RL11/RL01,RL02 DISK TYPES ON A VAX/VMS SYSTEM.  
0000 74  
0000 75 THE DISKS HAVE THE FOLLOWING PHYSICAL GEOMETRY:  
0000 76  
0000 77 # CYL TRACKS/ CYLINDER SECTORS/ TRACK BYTES/ SECTOR MAXIMUM  
0000 78 79 BLOCKS  
0000 80 RL01 256 2 40 256 10240  
0000 81 RL02 512 2 40 256 20480  
0000 82  
0000 83 SINCE THE SECTOR SIZE IS ONLY 1/2 BLOCK, LOGICAL TO PHYSICAL  
0000 84 CONVERSION OF THE DISK ADDRESS IS DONE IN THE DRIVER STARTIO  
0000 85 ROUTINE RATHER THAN IN THE IOCSVTLOGPHY FDT ROUTINE.  
0000 86  
0000 87 OVERLAPPED SEEKS ARE NOT ATTEMPTED BECAUSE THE DEVICE DOES  
0000 88 NOT INTERRUPT AT THE COMPLETION OF A SEEK.  
0000 89  
0000 90 ALSO, THE DEVICE DOES NOT PERFORM AN IMPLICIT SEEK WHEN PERFORMING  
0000 91 A READ OR WRITE FUNCTION, SO SEEK FUNCTIONS ARE ISSUED BY THIS  
0000 92 DRIVER WHERE NECESSARY PRIOR TO ISSUING A READ OR WRITE FUNCTION.  
0000 93 THE READ OR WRITE FUNCTION IS THEN ISSUED AS SOON AS THE RL11  
0000 94 CONTROLLER COMES READY (WHILE THE SEEK IS IN PROGRESS), AND A  
0000 95 WAIT FOR INTERRUPT (UPON COMPLETION OF THE READ OR WRITE) IS  
0000 96 ISSUED. IF A SEEK FUNCTION IS REQUESTED SEPARATELY FROM A READ OR  
0000 97 WRITE, A DUMMY READ HEADER FUNCTION IS ISSUED FOLLOWING THE SEEK  
0000 98 FUNCTION AND A WAIT FOR INTERRUPT (UPON COMPLETION OF THE READ  
0000 99 HEADER) IS ISSUED.  
0000 100  
0000 101 THE IOSX INHSEEK FUNCTION MODIFIER IS TREATED AS A NO-OP BY  
0000 102 THIS DRIVER, SINCE AN EXPLICIT SEEK IS NECESSARY FOR THE RL02  
0000 103 TO TRANSFER DATA PROPERLY.  
0000 104  
0000 105 THE RL'S DO NOT READ OR WRITE BEYOND THE END OF TRACK (THEY DO NOT  
0000 106 AUTOMATICALLY SEEK THE NEXT TRACK), SO ALL READ AND WRITE FUNCTIONS  
0000 107 ARE BROKEN UP BY THIS DRIVER INTO PARTIAL TRANSFERS TO THE END OF  
0000 108 TRACK, FOLLOWED BY A SEEK TO THE NEXT TRACK, THEN ANOTHER READ OR  
0000 109 WRITE FUNCTION UNTIL THE TOTAL DATA TRANSFER IS COMPLETE.  
0000 110  
0000 111 :--

```
0000 113 .SBTLL EXTERNAL AND LOCAL DEFINITIONS
0000 114
0000 115 :
0000 116 : EXTERNAL SYMBOLS
0000 117 :
0000 118
0000 119     $ADPDEF          ;DEFINE ADAPTER CONTROL BLOCK
0000 120     $CRBDEF          ;DEFINE CHANNEL REQUEST BLOCK
0000 121     $DCDEF            ;DEFINE DEVICE CLASS
0000 122     $DDBDEF          ;DEFINE DEVICE DATA BLOCK
0000 123     $DEVDEF           ;DEFINE DEVICE CHARACTERISTICS
0000 124     $DPTDEF           ;DEFINE DRIVER PROLOGUE TABLE
0000 125     $DYNDEF            ;DEFINE DYNAMIC DATA STRUCTURE TYPES
0000 126     $SEBDEF            ;DEFINE ERROR MESSAGE BUFFER
0000 127     $IDBDEF            ;DEFINE INTERRUPT DATA BLOCK
0000 128     $IODEF             ;DEFINE I/O FUNCTION CODES
0000 129     $IRPDEF            ;DEFINE I/O REQUEST PACKET
0000 130     $PRDEF             ;DEFINE PROCESSOR REGISTERS
0000 131     $PTEDEF            ;DEFINE SYSTEM PTES
0000 132     $SSDEF              ;DEFINE SYSTEM STATUS CODES
0000 133     $UCBDEF            ;DEFINE UNIT CONTROL BLOCK
0000 134     $VADEF              ;DEFINE VIRTUAL ADDRESS BITS
0000 135     $VECDEF            ;DEFINE INTERRUPT VECTOR BLOCK
0000 136
0000 137 :
0000 138 : LOCAL MACROS
0000 139 :
0000 140
0000 141 : EXFUNCL
0000 142 : BRANCH TO SUBROUTINE WHICH REQUESTS CHANNEL (IF NOT ALREADY OWNED),
0000 143 : EXECUTES FCODE (OR R3) FUNCTION, AND BRANCHES TO BDST ON ERROR
0000 144
0000 145     .MACRO EXFUNCL BDST,FCODE
0000 146         .IF NB FCODE          ;IS FCODE NON-BLANK?
0000 147         MOVZBL #CD'FCODE,R3   ;IF NB - SPECIFY FCODE FUNCTION
0000 148         .ENDC               ;IF B - SPECIFY FNTN IN EXISTING R3
0000 149         BSBW    FEXL          ;EXECUTE FUNCTION
0000 150         .BYTE    BDST-.,-1    ;WHERE TO GO IF ERROR
0000 151     .ENDM
0000 152
0000 153
0000 154 : GENF
0000 155 : GENERATE FUNCTION TABLE ENTRY AND CASE TABLE INDEX SYMBOL
0000 156
0000 157     .MACRO GENF FCODE
0000 158         CD'FCODE=.-FTAB/2
0000 159         .WORD    FCODE!RL_CS_M_IE ;FCODE WITH INT ENABLE BIT
0000 160     .ENDM
0000 161
0000 162
0000 163 : CKPWR
0000 164 : DISABLE INTERRUPTS, CHECK IF POWER HAS FAILED,
0000 165 : AND PUT DEVICE UNIT NUMBER IN R2<9:8>
0000 166 :
0000 167
0000 168     .MACRO CKPWR ?L1
0000 169     CLRL    R2          ;CLEAR R2 FOR UNIT NUMBER
```

```

0000 170      INSV    UCBSW_UNIT(R5),- ;PUT UNIT # IN R2<9:8>
0000 171
0000 172      DSBINT UCBSV_POWER - ;DISABLE INTERRUPTS
0000 173      BBC     #UCBSV_POWER - ;IF CLR - NO POWER FAILURE
0000 174
0000 175      ENBINT UCBSW_STS(R5),L1 ;POWER FAILURE - ENABLE INTERRUPTS
0000 176      BRW     RETREG   :EXIT
0000 177 L1:    .ENDM   :RETURN FOR NO POWER FAILURE
0000 178
0000 179
0000 180
0000 181      : LOCAL SYMBOLS
0000 182
0000 183
0000 184
00000004 0000 185 RL_NUM_REGS =4      :NUMBER OF DEVICE REGISTERS
00000005 0000 186 RL_SLM =5      :STATE=SEEK LINEAR MODE (READY TO GO)
000000C9 0000 187 UCBSB_DL_DCHEK =UCBSW_OFFSET+1 :REDEFINE FOR DATA CHECK USE
0000 188
0000 189      : UCB OFFSETS WHICH FOLLOW THE STANDARD UCB FIELDS
0000 190
0000 191      : SDEFINI UCB          :START OF UCB DEFINITIONS
0000 192
0000 193
000000CC 0000 194 .=UCBSK_LCL_DISK_LENGTH :BEGIN DEFINITIONS AT END OF UCB
0000 195 $DEF  UCBSW_DL_PBCR .BLKW 1 :PARTIAL BYTE COUNT
000E 196 $DEF  UCBSW_DL_CS .BLKW 1 :CONTROL STATUS REGISTER
0000 197 $DEF  UCBSW_DL_BA .BLKW 1 :BUS ADDRESS REGISTER
0002 198 $DEF  UCBSW_DL_DA .BLKW 1 :DISK ADDRESS REGISTER
0004 199 $DEF  UCBSW_DL_MP .BLKW 1 :MULTIPURPOSE REGISTER
0006 200 $DEF  UCBSW_DL_DPN .BLKW 1 :DATA PATH NUMBER
0008 201 $DEF  UCBSL_DL_SVAPTE .BLKL 1 :SAVED SVAPTE OF THE USER'S BUFFER
0008 202 $DEF  UCBSL_DL_DPR .BLKL 1 :DATAPATH REGISTER
000C 203 $DEF  UCBSL_DL_BUFADR .BLKL 1 :USER BUFFER ADDRESS
000C 204 $DEF  UCBSL_DL_FMPR .BLKL 1 :FINAL MAP REGISTER
00E0 205 $DEF  UCBSA_DL_MOVRN .BLKL 1 :BUFFER MOVE ROUTINE ADDRESS
00E0 206 $DEF  UCBSL_DL_PMPR .BLKL 1 :PREVIOUS MAP REGISTER
00E4 207 $DEF  UCBSB_DL_DPPE .BLKB 1 :DATAPATH PURGE ERROR
00E5 208 $DEF  UCBSW_DL_DB .BLKW 3 :DATA BUFFER REGISTER
00EB 209 $DEF  UCBSB_DL_XBA .BLKB 1 :BUS ADDRESS EXTENSION BITS
00EC 210 $DEF  UCBSW_DL_SBA .BLKW 1 :SAVED BUFFER ADDRESS
00EE 211 $DEF  UCBSA_DL_BUF_VA .BLKL 1 :PHYSICAL BUFFER VIRTUAL ADDRESS
00F2 212 $DEF  UCBSA_DL_BUF_PA .BLKL 1 :PHYSICAL BUFFER PHYSICAL ADDRESS
00F6 213 $DEF  UCBSW_DL_FLAGS .BLKW 1 :FLAGS
00FB 214 $YIELD OCB,0,<-          :START THE FLAG DEFINITIONS
00FB 215             <DL_22BIT,,M>,- :22 BIT ADDRESSING
00FB 216             <DL_MAPPING,,M>,- :ADAPTER MAPPING
00FB 217             >          :END OF FLAG DEFINITIONS
00FB 218 $DEF  UCBSK_DL_LEN .BLKW 1 :LENGTH OF UCB
00FA 219 SEQU UCBSK_DL_BUFSZ 20 :BUFFER SIZE = 40 SECTORS *
00FA 220
00FA 221      SDEFEND UCB :256 BYTES/SECTOR / 512 BYTES/PAGE
0000 222
0000 223
0000 224      : RL11/RL01 REGISTER OFFSETS FROM CSR ADDRESS
0000 225
0000 226      SDEFINI RL          : START OF REGISTER DEFINITIONS

```

0000	227						
0000	228	SDEF	RL_CS	.BLKW	1	:CONTROL STATUS REGISTER (CSR)	
0002	229	_VIELD	RL_CS,0,<-			:START OF CSR BIT DEFINITIONS	
0002	230		<DRDY,,M>,-			: DRIVE READY	
0002	231		<FCODE,3>,-			: FUNCTION CODE	
0002	232		<XBA,2>,-			: BUS ADDRESS EXTENSION BITS	
0002	233		<IE,,M>,-			: INTERRUPT ENABLE	
0002	234		<CRDY,,M>,-			: CONTROLLER READY	
0002	235		<DS2>,-			: DRIVE SELECT	
0002	236		<COP1,,M>,-			: OPERATION INCOMPLETE	
0002	237		<CRC,,M>,-			: DATA CRC OR HEADER CRC	
0002	238		<DLT,,M>,-			: DATA LATE OR HEADER NOT FOUND	
0002	239		<NXM,,M>,-			: NON-EXISTENT MEMORY	
0002	240		<DE,,M>,-			: DRIVE ERROR	
0002	241		<CE,,M>,-			: COMPOSITE ERROR	
0002	242		>			:END CSR BIT DEFINITIONS	
0002	243						
0002	244	SDEF	RL_BA	.BLKW	1	:BUS ADDRESS REGISTER (BAR)	
0004	245						
0004	246	SDEF	RL_DA	.BLKW	1	:DISK ADDRESS REGISTER (DAR)	
0006	247	_VIELD	RL_DA,0,<-			:START OF DAR BIT DEFINITIONS	
0006	248		<MRK,,M>,-			: MARK (ALWAYS 1)	
0006	249		<STS,,M>,-			: GET STATUS	
0006	250		<,1>,-			: RESERVED BIT	
0006	251		<RST,,M>,-			: RESET	
0006	252		<,12>,-			: RESERVED BITS	
0006	253		>			:END OF DAR BIT DEFINITIONS	
0006	254						
0006	255	SDEF	RL_MP	.BLKW	1	:MULTIPURPOSE REGISTER (MPR)	
0008	256	_VIELD	RL_MP,0,<-			:START OF MPR BIT DEFINITIONS	
0008	257		<STA,5>,-			: DRIVE STATE	
0008	258		<BH,,M>,-			: BRUSH HOME	
0008	259		<HO,,M>,-			: HEADS OUT	
0008	260		<CO,,M>,-			: COVER OPEN	
0008	261		<HS,,M>,-			: HEAD SELECT	
0008	262		<TYP,,M>,-			: DRIVE TYPE	
0008	263		<DSE,,M>,-			: DRIVE SELECT ERROR	
0008	264		<VVC,,M>,-			: VOLUME CHECK	
0008	265		<WGE,,M>,-			: WRITE GATE ERROR	
0008	266		<SPE,,M>,-			: SPIN ERROR	
0008	267		<SKTO,,M>,-			: SEEK TIME OUT	
0008	268		<WL,,M>,-			: WRITE LOCK	
0008	269		<CHÉ,,M>,-			: CURRENT HEAD ERROR	
0008	270		<WDE,,M>,-			: WRITE DATA ERROR	
0008	271		>			:END MPR BIT DEFINITIONS	
0008	272						
0008	273	SDEF	RL_BAE	.BLKW	1	: BUS ADDRESS EXTENSION REGISTER(BAE)	
000A	274						
000A	275	SDEFEND	RL			:END RL11/RL01 REGISTER DEFINITIONS	
0000	276						
0000	277						
0000	278		: HARDWARE FUNCTION CODES				
00000000	279						
00000000	280	F NOP=0*2				:NO OPERATION	
00000000	281	F UNLOAD=F NOP				:NO OPERATION	
00000006	282	F SEEK=3*2				:SEEK CYLINDER	
00000000	283	F RECAL=F NOP				:NO OPERATION	

00000004	0000	284	F_DRVCLR=2*2	:DRIVE CLEAR (GET STATUS)
00000000	0000	285	F_RELEASE=F NOP	:NO OPERATION
00000000	0000	286	F_OFFSET=F NOP	:NO OPERATION
00000000	0000	287	F_RETCENTER=F NOP	:NO OPERATION
00000004	0000	288	F_PACKACK=2*2	:PACK ACKNOWLEDGE (SET VOLUME VALID)
00000000	0000	289	F_SEARCH=F NOP	:NO OPERATION
00000002	0000	290	F_WRITECHECK=1*2	:WRITE CHECK
0000000A	0000	291	F_Writedata=5*2	:WRITE DATA
00000000	0000	292	F_WRITEHEAD=F NOP	:NO OPERATION
0000000C	0000	293	F_READDATA=6*2	:READ DATA
00000008	0000	294	F_READHEAD=4*2	:READ HEADER
00000000	0000	295	F_AVAILABLE=F NOP	:NO OPERATION
00000004	0000	296	F_GETSTATUS=2*2	:GET STATUS (DRIVER INTERNAL USE)
		297		

0000 299 .SBTTL STANDARD TABLES

0000 300

0000 301 :: DRIVER PROLOGUE TABLE

0000 302 ::

0000 303 :: THE DPT DESCRIBES DRIVER PARAMETERS AND I/O DATABASE FIELDS

0000 304 :: THAT ARE TO BE INITIALIZED DURING DRIVER LOADING AND RELOADING

0000 305 ::

0000 306 ::

0000 307 ::

0000 308 DPTAB - :DPT CREATION MACRO

0000 309 END=DL END,- :END OF DRIVER LABEL

0000 310 ADAPTER=UBA,- :ADAPTER TYPE = UNIBUS

0000 311 FLAGS=DPTSM\_SVP,- :SYSTEM PAGE TABLE ENTRY REQUIRED

0000 312 UCBSIZE=UCBSK\_DL\_LEN,- :LENGTH OF UCB

0000 313 NAME=DLDRIVER :DRIVER NAME

0038 314

0038 315 DPT\_STORE INIT :START CONTROL BLOCK INIT VALUES

0038 316 DPT\_STORE DDB,DDB\$L\_ACPD,L,<"A\FL1> :DEFAULT ACP NAME

003F 317 DPT\_STORE DDB,DDB\$L\_ACPD+3,B,DDBSK\_CART :ACP CLASS

0043 318 DPT\_STORE UCB,UCBSB\_FIPL,B,8 :FORK IPL

0047 319 DPT\_STORE UCB,UCBSL\_DEVCHAR,L,- :DEVICE CHARACTERISTICS

0047 320 <DEVSM\_FOD- :FILES ORIENTED

0047 321 !DEVSM\_DIR- :DIRECTORY STRUCTURED

0047 322 !DEVSM\_AVL- :AVAILABLE

0047 323 !DEVSM\_ELG- :ERROR LOGGING

0047 324 !DEVSM\_SHR- :SHAREABLE

0047 325 !DEVSM\_IDV- :INPUT DEVICE

0047 326 !DEVSM\_ODV- :OUTPUT DEVICE

0047 327 !DEVSM\_RND> :RANDOM ACCESS

004E 328 DPT\_STORE UCB,UCBSL\_DEVCHAR2,L,- :DEVICE CHARACTERISTICS

004E 329 <DEVSM\_NNM> :PREFIX NAME WITH "nodes"

0055 330 DPT\_STORE UCB,UCBSB\_DEVCLASS,B,DCS\_DISK :DEVICE CLASS

0059 331 DPT\_STORE UCB,UCBSB\_DEVBUFSIZ,W,512 :DEFAULT BUFFER SIZE

005E 332 DPT\_STORE UCB,UCBSB\_SECTORS,B,40 :NUMBER OF SECTORS PER TRACK

0062 333 DPT\_STORE UCB,UCBSB\_TRACKS,B,2 :NUMBER OF TRACKS PER CYLINDER

0066 334 DPT\_STORE UCB,UCBSB\_DIPL,B,21 :DEVICE IPL

006A 335 DPT\_STORE UCB,UCBSB\_ERTMAX,B,8 :MAX ERROR RETRY COUNT

006E 336 DPT\_STORE UCB,UCBSW\_DEVSTS,W,- :INHIBIT LOG TO PHYS CONVERSION IN FDT

006E 337 <UCBSM\_NOCNVRT> :....

0073 338

0073 339 DPT\_STORE REINIT :START CONTROL BLOCK RE-INIT VALUES

0073 340 DPT\_STORE CRB,CRB\$L\_INTD+4,D,DL INT :INTERRUPT SERVICE ROUTINE ADDRESS

0078 341 DPT\_STORE CRB,CRB\$L\_INTD+VEC\$L\_INITIAL,- :CONTROLLER INIT ADDRESS

007D 342 D,DL RL11\_INIT

007D 343 DPT\_STORE CRB,CRB\$C\_INTC+VEC\$L\_UNITINIT,- :UNIT INIT ADDRESS

007D 344 D,DL RLOX\_INIT

0082 345 DPT\_STORE DDB,DDB\$C\_DDT,D,DL\$DDT :DDT ADDRESS

0087 346

0087 347 DPT\_STORE END :END OF INITIALIZATION TABLE

0000 348

0000 349 :: DRIVER DISPATCH TABLE

0000 350 ::

0000 351 ::

0000 352 :: THE DDT LISTS ENTRY POINTS FOR DRIVER SUBROUTINES WHICH ARE

0000 353 :: CALLED BY THE OPERATING SYSTEM.

0000 354 ::

0000 355 ::

0000	356	DDTAB	-	: DDT CREATION MACRO
0000	357		DEVNAME=DL,-	: NAME OF DEVICE
0000	358		START=DL \$STARTIO,-	: START I/O ROUTINE
0000	359		UNSOLIC=BL UNSOLNT,-	: UNSOLICITED INTERRUPT
0000	360		FUNCTB=DL FUNCTABLE,-	: FUNCTION DECISION TABLE
0000	361		CANCEL=0,-	: CANCEL=NO-OP FOR FILES DEVICE
0000	362		REGDMP=DL REGDUMP,-	: REGISTER DUMP ROUTINE
0000	363		DIAGBF=<<RL_NUM_REGS+5+5+3+1>>*4> -	: BYTES IN DIAG BUFFER
0000	364		ERLGBF=<<RE_NUM_REGS+5+1>>*4>+EMBSL_DV_REGSAV>	: BYTES IN
0038	365			: ERROR LOG BUFFER
0038	366			
0038	367		: DIAGNOSTIC BUFFER SIZE = <<4 RL02 REGISTER LONGWORDS + 5 UCB FIELD LONGWORDS	
0038	368		+ 5 IOCS\$DIAGBUFL LONGWORDS + 3 BUFFER ALLOCATION	
0038	369		LONGWORDS + 1 LONGWORD FOR # REGISTERS IN DL_REGDUMP>	
0038	370		* 4 BYTES/LONGWORD>	
0038	371			
0038	372		: ERROR LOG BUFFER SIZE = <<<4 RL02 REGISTER LONGWORDS + 5 UCB FIELD LONGWORDS	
0038	373		+ 1 LONGWORD FOR # REGISTERS IN DL_REGDUMP>	
0038	374		* 4 BYTES/LONGWORD> + BYTES NEEDED FOR ERROR LOGGER	
0038	375		TO SAVE SOFTWARE REGISTERS>	
0038	376			
0038	377			
0038	378			
0038	379		: HARDWARE FUNCTION CODE TABLE	
0038	380			
0038	381		THIS TABLE MERGES THE FUNCTION CODE BITS WITH THE	
0038	382		INTERRUPT ENABLE BIT AND GENERATES THE CASE TABLE	
0038	383		INDEX SYMBOL.	
0038	384			
0038	385	FTAB:	GENF F NOP	: NO-OP
003A	386		GENF F UNLOAD	: UNLOAD VOLUME (NOP)
003C	387		GENF F SEEK	: SEEK
003E	388		GENF F RECAL	: RECALIBRATE (NOP)
0040	389		GENF F DRVCLR	: DRIVE CLEAR (RESET & GET STATUS)
0042	390		GENF F RELEASE	: RELEASE PORT (NOP)
0044	391		GENF F OFFSET	: OFFSET HEADS (NOP)
0046	392		GENF F RETCENTER	: RETURN HEADS TO CENTERLINE (NOP)
0048	393		GENF F PACKACK	: PACK ACKNOWLEDGE (RESET & GET STATUS)
004A	394		GENF F SEARCH	: SEARCH (NOP)
004C	395		GENF F WRITECHECK	: WRITE CHECK
004E	396		GENF F WRITEDATA	: WRITE DATA
0050	397		GENF F READDATA	: READ DATA
0052	398		GENF F WRITEHEAD	: WRITE HEADERS (NOP)
0054	399		GENF F READHEAD	: READ HEADERS
0056	400		GENF F NOP	: place holder
0058	401		GENF F NOP	: place holder
005A	402		GENF F AVAILABLE	: AVAILABLE
005C	403			

005C 405	:	FUNCTION DECISION TABLE	
005C 406	:	THE FDT LISTS VALID FUNCTION CODES, SPECIFIES WHICH	
005C 407	:	CODES ARE BUFFERED, AND DESIGNATES SUBROUTINES TO	
005C 408	:	PERFORM PREPROCESSING FOR PARTICULAR FUNCTIONS.	
005C 409	:		
005C 410	:		
005C 411	:		
005C 412	:		
005C 413	DL_FUNCTABLE:		
005C 414	FUNCTAB , -		: LIST LEGAL FUNCTIONS
005C 415	<NOP,-		: NO-OP
005C 416	UNLOAD,-		: UNLOAD
005C 417	SEEK,-		: SEEK
005C 418	DRVCLR,-		: DRIVE CLEAR
005C 419	PACKACK,-		: PACK ACKNOWLEDGE
005C 420	SENSECHAR,-		: SENSE CHARACTERISTICS
005C 421	SETCHAR,-		: SET CHARACTERISTICS
005C 422	SENSEMODE,-		: SENSE MODE
005C 423	SETMODE,-		: SET MODE
005C 424	WRITECHECK,-		: WRITE CHECK
005C 425	READHEAD,-		: READ HEADER
005C 426	READLBLK,-		: READ LOGICAL BLOCK
005C 427	WRITELBLK,-		: WRITE LOGICAL BLOCK
005C 428	READPBLK,-		: READ PHYSICAL BLOCK
005C 429	WRITEPBLK,-		: WRITE PHYSICAL BLOCK
005C 430	READVBLK,-		: READ VIRTUAL BLOCK
005C 431	WRITEVBLK,-		: WRITE VIRTUAL BLOCK
005C 432	AVAILABLE,-		: AVAILABLE
005C 433	ACCESS,-		: ACCESS FILE / FIND DIRECTORY ENTRY
005C 434	ACPCONTROL,-		: ACP CONTROL FUNCTION
005C 435	CREATE,-		: CREATE FILE AND/OR DIRECTORY ENTRY
005C 436	DEACCESS,-		: DEACCESS FILE
005C 437	DELETE,-		: DELETE FILE AND/OR DIRECTORY ENTRY
005C 438	MODIFY,-		: MODIFY FILE ATTRIBUTES
005C 439	MOUNT-		: MOUNT VOLUME
005C 440	>		
0064 441	FUNCTAB , -		: BUFFERED FUNCTIONS
0064 442	<NOP,-		: NO-OP
0064 443	UNLOAD,-		: UNLOAD
0064 444	SEEK,-		: SEEK
0064 445	DRVCLR,-		: DRIVE CLEAR
0064 446	PACKACK,-		: PACK ACKNOWLEDGE
0064 447	SENSECHAR,-		: SENSE CHARACTERISTICS
0064 448	SETCHAR,-		: SET CHARACTERISTICS
0064 449	SENSEMODE,-		: SENSE MODE
0064 450	SETMODE,-		: SET MODE
0064 451	AVAILABLE,-		: AVAILABLE
0064 452	ACCESS,-		: ACCESS FILE / FIND DIRECTORY ENTRY
0064 453	ACPCONTROL,-		: ACP CONTROL FUNCTION
0064 454	CREATE,-		: CREATE FILE AND/OR DIRECTORY ENTRY
0064 455	DEACCESS,-		: DEACCESS FILE
0064 456	DELETE,-		: DELETE FILE AND/OR DIRECTORY ENTRY
0064 457	MODIFY,-		: MODIFY FILE ATTRIBUTES
0064 458	MOUNT-		: MOUNT VOLUME
0064 459	>		
006C 460	FUNCTAB DL ALIGN,-		: TEST ALIGNMENT FUNCTIONS
006C 461	<READHEAD,-		: READ HEADER

006C	462	READBLK,-	: READ LOGICAL BLOCK
006C	463	READPBLK,-	: READ PHYSICAL BLOCK
006C	464	READVBLK,-	: READ VIRTUAL BLOCK
006C	465	WRITECHECK,-	: WRITE CHECK
006C	466	WRITELBLK,-	: WRITE LOGICAL BLOCK
006C	467	WRITEPBLK,-	: WRITE PHYSICAL BLOCK
006C	468	WRITEVBLK-	: WRITE VIRTUAL BLOCK
006C	469	>	
0078	470	FUNCTAB +ACPSREADBLK,-	: READ FUNCTIONS
0078	471	<READHEAD,-	: READ HEADER
0078	472	READLBLK,-	: READ LOGICAL BLOCK
0078	473	READPBLK,-	: READ PHYSICAL BLOCK
0078	474	READVBLK-	: READ VIRTUAL BLOCK
0078	475	>	
0084	476	FUNCTAB +ACPSWRITEBLK,-	: WRITE FUNCTIONS
0084	477	<WRITECHECK,-	: WRITE CHECK
0084	478	WRITELBLK,-	: WRITE LOGICAL BLOCK
0084	479	WRITEPBLK,-	: WRITE PHYSICAL BLOCK
0084	480	WRITEVBLK-	: WRITE VIRTUAL BLOCK
0084	481	>	
0090	482	FUNCTAB +ACPSACCESS,-	: ACCESS FUNCTIONS
0090	483	<ACCESS,-	: ACCESS FILE / FIND DIRECTORY ENTRY
0090	484	CREATE-	: CREATE FILE AND/OR DIRECTORY ENTRY
0090	485	>	
009C	486	FUNCTAB +ACPSDEACCESS,-	: DEACCESS FUNCTION
009C	487	<DEACCESS-	: DEACCESS FILE
009C	488	>	
00A8	489	FUNCTAB +ACPSMODIFY,-	: MODIFY FUNCTIONS
00A8	490	<ACPCONTROL,-	: ACP CONTROL FUNCTION
00A8	491	DELETE,-	: DELETE FILE AND/OR DIRECTORY ENTRY
00A8	492	MODIFY-	: MODIFY FILE ATTRIBUTES
00A8	493	>	
00B4	494	FUNCTAB +ACPSMOUNT,-	: MOUNT FUNCTION
00B4	495	<MOUNT-	: MOUNT VOLUME
00B4	496	>	
00C0	497	FUNCTAB +EXESLCLDISKVALID,-	: LOCAL DISK VALID FUNCTIONS
00C0	498	<UNLOAD,-	: UNLOAD VOLUME
00C0	499	AVAILABLE,-	: UNIT AVAILABLE
00C0	500	PACKACK-	: PACK ACKNOWLEDGE
00C0	501	>	
00CC	502	FUNCTAB +EXESZEROOPARM,-	: ZERO PARAMETER FUNCTIONS
00CC	503	<NOP,-	: NO-OP
00CC	504	UNLOAD,-	: UNLOAD
00CC	505	DRVCLR,-	: DRIVE CLEAR
00CC	506	PACKACK,-	: PACK ACKNOWLEDGE
00CC	507	AVAILABLE,-	: AVAILABLE
00CC	508	>	
00D8	509	FUNCTAB +EXESONEOPARM,-	: ONE PARAMETER FUNCTION
00D8	510	<SEEK-	: SEEK
00D8	511	>	
00E4	512	FUNCTAB +EXESSENSEMODE,-	: SENSE FUNCTIONS
00E4	513	<SENSECHAR,-	: SENSE CHARACTERISTICS
00E4	514	SENSEMODE-	: SENSE MODE
00E4	515	>	
00F0	516	FUNCTAB +EXESSETCHAR,-	: SET FUNCTIONS
00F0	517	<SETCHAR,-	: SET CHARACTERISTICS
00F0	518	SETHOOK-	: SET MODE

DLDRI<sup>E</sup>  
V04-000

- VAX/VMS RL11/RL01,RL02 DISK DRIVER  
STANDARD TABLES

F 6

00F0 519

>

16-SEP-1984 00:17:29 VAX/VMS Macro V04-00  
5-SEP-1984 00:12:24 [DRIVER.SRC]DLDRI<sup>E</sup>.MAR;1 Page 12  
(1)

DL  
VC

00FC 521 .SBTTL CONTROLLER INITIALIZATION ROUTINE

00FC 522 \*\*

00FC 523 FUNCTIONAL DESCRIPTION:

00FC 524 THIS ROUTINE IS A NO-OP FOR THE RL11 BUT MUST BE INCLUDED  
00FC 525 SINCE IT IS CALLED WHEN THE RL02 IS BOOTTED AS A SYSTEM DEVICE.

00FC 526 THE OPERATING SYSTEM CALLS THIS ROUTINE:

00FC 527 - AT SYSTEM STARTUP  
00FC 528 - DURING DRIVER LOADING  
00FC 529 - DURING RECOVERY FROM POWER FAILURE

00FC 530 INPUTS:

00FC 531 R4 - CSR ADDRESS (DEVICE CONTROL STATUS REGISTER)  
00FC 532 R5 - IDB ADDRESS (INTERRUPT DATA BLOCK)  
00FC 533 R6 - DDB ADDRESS (DEVICE DATA BLOCK)  
00FC 534 R8 - CRB ADDRESS (CHANNEL REQUEST BLOCK)  
00FC 535 ALL INTERRUPTS ARE LOCKED OUT

00FC 536 OUTPUTS:

00FC 537 ALL REGISTERS EXCEPT R0-R3 ARE PRESERVED.  
00FC 538 CONTROL IS RETURNED TO THE CALLER.

00FC 539 :--

00FC 540 DL\_RL11\_INIT: ;CONTROLLER INITIALIZATION

00FC 541 ; FOR MICROVAX I, ALLOCATE A PHYSICALLY CONTIGUOUS BUFFER

00FC 542 ; AREA FOR PERFORMING I/O.

00FC 543 ;(PUDISP <>790,20\$>,-  
00FC 544 <785,20\$>,-  
00FC 545 <780,20\$>,-  
00FC 546 <750,20\$>,-  
00FC 547 <730,20\$>,-  
00FC 548 <UV1,10\$> ;FOR MICROVAX I, ALLOCATE BUFFER AREA  
012E 549 ;FOR ALL OTHERS, SKIP BUFFER AREA

012E 550 ;MOVZUL #UCBSK DL\_BUFSZ,R1 ;LOAD SIZE OF BUFFER  
012E 551 JSB G\$EXES\$ALOPHYCNTG ;ALLOCATE PHYSICALLY-CONTIGUOUS MEMORY  
012E 552 BLBC R0,20\$ ;EXIT ON ERROR  
012E 553 MOVL R2,CRBSL\_AUXSTRUC(R8) ;GET BUFFER VIRTUAL ADDRESS  
012E 554 RSB ;RETURN TO CALLER

51 14 3C 012E 555 10\$: CLRL CRBSL\_AUXSTRUC(R8) ;INDICATE MEMORY ALLOCATION FAILURE  
00000000 GF 16 0131 556 RSB ;RETURN TO CALLER  
05 50 E9 0137 557  
10 AB S2 D0 013A 558  
05 013E 559  
013F 560  
013F 561  
012E 562 ;

10 AB D4 013F 563 10\$: MOVZUL #UCBSK DL\_BUFSZ,R1 ;LOAD SIZE OF BUFFER  
05 0142 564 JSB G\$EXES\$ALOPHYCNTG ;ALLOCATE PHYSICALLY-CONTIGUOUS MEMORY  
570 20\$: CLRL CRBSL\_AUXSTRUC(R8) ;INDICATE MEMORY ALLOCATION FAILURE  
RSB ;RETURN TO CALLER

0163 572 .SBTTL UNIT INITIALIZATION ROUTINE  
 0164 573  
 0164 574 ;\*\*  
 0164 575 : DL\_RLOX\_INIT - UNIT INITIALIZATION ROUTINE  
 0164 576  
 0164 577 : FUNCTIONAL DESCRIPTION:  
 0164 578  
 0164 580 : THIS ROUTINE READIES THE RL01/RL02 UNITS FOR I/O OPERATIONS.  
 0164 581  
 0164 582 : THE OPERATING SYSTEM CALLS THIS ROUTINE:  
 0164 583 : - AT SYSTEM STARTUP  
 0164 584 : - DURING DRIVER LOADING  
 0164 585 : - DURING RECOVERY FROM POWER FAILURE  
 0164 586  
 0164 587 : INPUTS:  
 0164 588  
 0164 589 R4 : - CSR ADDRESS (CONTROLLER STATUS REGISTER)  
 0164 590 R5 : - UCB ADDRESS (UNIT CONTROL BLOCK)  
 0164 591  
 0164 592 : OUTPUTS:  
 0164 593  
 0164 594 : THE DRIVE UNIT IS RESET, UCB FIELDS ARE INITIALIZED, AND THE  
 0164 595 : ROUTINE WAITS FOR ONLINE UNITS TO SPIN UP. ALL REGISTERS  
 0164 596 : EXCEPT R0-R3 ARE PRESERVED.  
 0164 597  
 0164 598 :--  
 0164 599  
 00F6 02 B0 600 DL\_RLOX\_INIT: : RL01/RL02 UNIT INITIALIZATION  
 00F6 C5 601 MOVW #1@UCBSV\_DL\_MAPPING,- : DEFAULT TO ADAPTER MAPPING  
 602 UCBSV\_DL\_FLAGS(R5) : AND 18 BIT ADDRESSING  
 0164 603  
 0164 604 : SET CPU DEPENDENT UCB FLAGS FOR DL  
 0164 605 CPUDISP <>790,10\$>,-  
 0164 606 <785,10\$>,-  
 0164 607 <780,10\$>,-  
 0164 608 <750,10\$>,-  
 0164 609 <730,10\$>,-  
 0164 610 <UV1,5\$>  
 0164 611  
 53 00F6 01 B0 612 5\$: MOVW #1@UCBSV\_DL\_22BIT,- : FOR MICROVAX I 22 BIT  
 017A 613 UCBSV\_DL\_FLAGS(R5) : ADDRESSING AND NO ADAPTER MAPPING  
 017C 614 10\$: MOVZWL UCBSV\_STS(R5),R3 : SAVE CURRENT UNIT STATUS  
 017F 615 BICW #UCBSM\_ONLINE!UCBSM\_VALID,- :ASSUME OFFLINE/INVALID  
 0810 8F AA 616 UCBSV\_STS(R5) :...  
 64 A5 617  
 0164 618 : WAIT FOR CONTROLLER (6 SECONDS MAX) IF CHANNEL IS BUSY WITH ANOTHER UNIT  
 0164 619  
 0164 620  
 0164 621  
 29 50 24 A5 D0 0189 622 MOVL UCBSL(CRB(R5),R0 : GET CRB ADDRESS  
 0E A0 00 E1 0180 623 BBC #CRBSV\_BSY(CRBSB\_MASK(R0))20\$ :IF CLEAR - CHANNEL NOT BUSY  
 0192 624 TIMEDWAIT TIME=600+1000,- :6 SECOND WAIT LOOP  
 0192 625 INS1=<TSB RL CS(R4)>,- :IS CONTROLLER READY  
 0192 626 INS2=<BLSS 15\$>,- :IF LSS - YES  
 0192 627 DONELBL=15\$ :LABEL TO EXIT WAIT LOOP  
 3B 50 E9 0188 628 BLBC R0,25\$ :TIME EXPIRED - EXIT

0186 629 :  
 0188 630 : GET CURRENT DRIVE STATUS AND RESET DRIVE  
 0188 631 :  
 0188 632 :  
 04 A4 0B B0 0188 633 208: MOVW #RL\_DA\_M\_RST!- :PUT RESET AND GET STATUS IN DAR  
 S1 08 08 54 51 D4 018F 634 : RL\_DA\_M\_STS!RL\_DA\_M\_MRK,RL\_DA(R4)  
 64 04 04 51 F0 01C1 635 : CLRL R1 :CLEAR R1 FOR UNIT NUMBER  
 0095 30 01CB 636 : INSV UCBSW\_UNIT(R5),#8,#8,R1 :GET UNIT NUMBER  
 64 95 01CE 637 : BISW3 R1,#F-GETSTATUS,RL\_CS(R4) :EXECUTE GET STATUS FUNCTION  
 24 18 01D0 638 : BSBW DL\_WAIT :WAIT FOR CONTROLLER  
 01D2 639 : TSTB RL\_CS(R4) : WAS CONTROLLER READY?  
 01D2 640 : BGEQ 258 : IF GEO - NO  
 01D2 641 :  
 01D2 642 :  
 01D2 643 : CLASSIFY DRIVE TYPE  
 01D2 644 :  
 01D2 645 :  
 2324C001 8F D0 01D2 646 : MOVL \$^X2324C001,- :  
 008C C5 01DB 647 : UCBSL MEDIA\_ID(R5) :SET MEDIA IDENT "DL RL01"  
 06 A4 0080 8F B3 01DB 648 : BITW #RL\_MP\_M\_TYP,RL\_MP(R4) :IS DRIVE TYPE = RL02?  
 15 12 01E1 649 : BNEQ 308 :IF NEQ - YES  
 09 90 01E3 650 : MOVW \$^#DTS\_RL01,- :  
 46 A5 0100 8F B0 01E7 651 : UCBSB\_DEVTYPE(R5) :SET RL01 DEVICE TYPE  
 0080 C5 2800 8F 3C 01ED 652 : MOVW #256,UCBSW\_CYLINDERS(R5) :SET NUMBER OF RL01 CYLINDERS  
 17 11 01F4 653 : MOVZUL #10240,UCBSL\_MAXBLOCK(R5) :SET MAX RL01 BLOCK NUMBER  
 BRB 654 : 408  
 01F6 655 :  
 6A 11 01F6 656 258: BRB 708 :BRANCH TO COMMON EXIT  
 01FB 657 :  
 0A 90 01FB 658 308: MOVW \$^#DTS\_RL02,- :  
 41 A5 0200 8F B0 01FC 659 : UCBSB\_DEVTYPE(R5) :SET RL02 DEVICE TYPE  
 0080 C5 5000 8F 3C 0202 660 : MOVW #512,UCBSW\_CYLINDERS(R5) :SET NUMBER OF RL02 CYLINDERS  
 008C C5 D6 0209 661 : MOVZUL #20480,UCBSL\_MAXBLOCK(R5) :SET MAX RL02 BLOCK NUMBER  
 16 53 0B E1 0200 662 : INCL UCBSL\_MEDIA\_ID(R5) :SET MEDIA IDENT "DL RL02"  
 0211 663 408: BBC #UCBSV\_VALID,R3,608 : Branch around wait for drive to spinup  
 0211 664 : if the drive did NOT have a VALID  
 0211 665 : volume on it before POWER failure.  
 0211 666 :  
 0211 667 :  
 0211 668 : INITIALIZE UCB FIELDS AND WAIT FOR ONLINE UNITS TO SPIN UP  
 0211 669 :  
 0211 670 :  
 64 01 B3 0211 671 458: BITW #RL\_CS\_M\_DRDY,RL\_CS(R4) : Is drive ready?  
 00000000 GF 12 0214 672 : BNEQ 508 :IF NEQ - YES  
 F2 50 E8 0216 673 : JSB G^EXESPWRTIMCHK :IS MAX TIME EXCEEDED?  
 06 11 021F 674 : BLBS R0\_458 :IF LBS - NO, STILL MORE TIME NEEDED  
 0221 675 : BRB 608 :POWER UP TIME EXCEEDED  
 64 A5 0800 8F A8 0221 676 508: BISW #UCBSM\_VALID,UCBSU\_STS(R5) ;SET UCB STATUS VOLUME VALID  
 00F6 C5 01 E0 0227 677 :  
 31 24 A5 D0 0229 680 : BBS #UCBSV\_DL\_MAPPING- :ADAPTER MAPPING?  
 52 10 A1 D0 022D 681 : UCBSU\_DL\_FLAGS(R5),658 :IF BS YES  
 28 13 0231 682 : MOVL UCBSL\_CRB(R5),R1 :GET CRB ADDRESS  
 52 00EE C5 52 00 0232 683 : MOVL CRBSL\_AUXSTRUC(R1),R2 :MEMORY ALLOC FAILURE DURING CTL INIT?  
 52 15 09 EF 023C 684 : BEQL 708 :IF EQL YES, LEAVE OFFLINE  
 EXTZV R? UCBSA\_DL\_BUF\_VA(R5) :SAVE BUFFER'S VIRTUAL ADDRESS  
 0VASS\_VPN,#VASS\_VPN,R2,R1 :GET VIRTUAL PAGE NUMBER OF BUFFER

50	00000000'GF	00	0241	686	MOVL	G'MMG\$GL_SPTBASE,R0	:GET BASE ADDRESS OF SPTS
51	50 6041	00	0248	687	MOVL	(R0)[R1],R0	:GET THE PTE CONTENTS
51	52 FFFFFE00 BF	CB	024C	688	BICL3	#"<<VASM BYTE>>,R2,R1	:GET BUFFER OFFSET (BA00-BA08)
51	0D 09 50	F0	0254	690	ASSUME	PTESS_PFA GE 13	
00F2 C5 51	DO 0259	691	INSV	R0,#9-#13,R1	:COPY BA09-BA21		
64 A5 10	A8 025E	692	658:	MOVL	R1,UCBSA_DL_BUF_PA(R5)	:SAVE PHYSICAL ADDRESS OF BUFFER	
05 0262	693	708:	BISW	#UCBSM_ONLINE,UCBSW_STS(R5)	:SET UCB STATUS VOLUME VALID		
			RSB				

0263 695 .SBTTL DRIVER SPECIFIC SUBROUTINES  
0264 696  
0264 697 DL\_WAIT - WAIT FOR CONTROLLER READY  
0264 698  
0264 699 INPUTS:  
0264 700 R4 - DEVICE CSR ADDRESS  
0264 701  
0264 702 FUNCTIONAL DESCRIPTION:  
0264 703  
0264 704 THIS ROUTINE IS CALLED FROM THE DRIVER UNIT INITIALIZATION ROUTINE  
0264 705 TO WAIT UNTIL THE RL11 CONTROLLER IS READY. TO PREVENT HANGING UP  
0264 706 AT HIGH IPL, A MAXIMUM OF 30 USEC ELAPSES BEFORE CONTROL IS  
0264 707 RETURNED TO THE CALLER.  
0264 708  
0264 709  
0264 710 DL\_WAIT: :WAIT FOR CONTROLLER READY  
7E 50 7D 0263 711 MOVO R0,-(SP) :SAVE R0, R1  
0266 712 DSBINT :DISABLE INTERRUPTS  
026C 713 TIMEWAIT #3,NRL\_CS\_M\_CRDY,RL CS(R4) W  
0291 714 ENBINT :ENABLE INTERRUPTS  
50 8E 7D 0294 715 MOVO (SP)+,R0 :RESTORE R0, R1  
05 0297 716 RSB :RETURN TO UNIT INIT OR STARTIO

.SBTTL FDT ROUTINE - TEST TRANSFER BYTE COUNT ALIGNMENT

:::

DL\_ALIGN - FDT ROUTINE TO TEST XFER BYTE COUNT

FUNCTIONAL DESCRIPTION:

THIS ROUTINE IS CALLED FROM THE FUNCTION DECISION TABLE DISPATCHER  
 TO CHECK THE BYTE COUNT PARAMETER SPECIFIED BY THE USER PROCESS  
 FOR AN EVEN NUMBER OF BYTES (WORD BOUNDARY).

INPUTS:

- R3 - IRP ADDRESS (I/O REQUEST PACKET)
- R4 - PCB ADDRESS (PROCESS CONTROL BLOCK)
- R5 - UCB ADDRESS (UNIT CONTROL BLOCK)
- R6 - CCB ADDRESS (CHANNEL CONTROL BLOCK)
- R7 - BIT NUMBER OF THE I/O FUNCTION CODE
- R8 - ADDRESS OF FDT TABLE ENTRY FOR THIS ROUTINE
- 4(AP) - ADDRESS OF FIRST FUNCTION DEPENDENT QIO PARAMETER

OUTPUTS:

IF THE QIO BYTE COUNT PARAMETER IS ODD, THE I/O OPERATION IS  
 TERMINATED WITH AN ERROR. IF IT IS EVEN, CONTROL IS RETURNED  
 TO THE FDT DISPATCHER.

--

DL\_ALIGN:

01 04 AC	E8	0298	748	DL_ALIGN:		:CHECK BYTE COUNT AT P1(AP)
50 034C BF	05	029C	749	BLBS	4(AP),108	:IF LBS - ODD BYTE COUNT
00000000'GF	3C	029D	750	RSB		:EVEN - RETURN TO CALLER

108: MOVZWL #SS\$ IVBUFLLEN.R0

JMP G^EXESABORTIO

:SET BUFFER ALIGNMENT STATUS  
 :ABORT I/O

02AB 754 .SBTTL START I/O ROUTINE  
 02AB 755  
 02AB 756 ++  
 02AB 757  
 02AB 758 DL\_STARTIO - START I/O ROUTINE  
 02AB 759  
 02AB 760 FUNCTIONAL DESCRIPTION:  
 02AB 761  
 02AB 762 THIS FORK PROCESS IS ENTERED FROM THE EXECUTIVE AFTER AN I/O REQUEST  
 02AB 763 PACKET HAS BEEN DEQUEUED, AND PERFORMS THE FOLLOWING:  
 02AB 764  
 02AB 765 - ACTIVATES THE DISK AFTER SETTING UCB FIELDS, OBTAINING  
 02AB 766 UBA AND CONTROLLER RESOURCES, AND SETTING RL11 REGISTERS  
 02AB 767  
 02AB 768  
 02AB 769  
 02AB 770  
 02AB 771  
 02AB 772  
 02AB 773  
 02AB 774  
 02AB 775  
 02AB 776 INPUTS:  
 02AB 777  
 02AB 778 R3 - IRP ADDRESS (I/O REQUEST PACKET)  
 02AB 779 R5 - UCB ADDRESS (UNIT CONTROL BLOCK)  
 02AB 780 IRPSL\_MEDIA - PARAMETER LONGWORD (LOGICAL BLOCK NUMBER)  
 02AB 781  
 02AB 782 OUTPUTS:  
 02AB 783  
 02AB 784 R0 - FIRST I/O STATUS LONGWORD: STATUS CODE & BYTES XFERRED  
 02AB 785 R1 - SECOND I/O STATUS LONGWORD: 0 FOR DISKS  
 02AB 786  
 02AB 787  
 02AB 788 THE I/O FUNCTION IS EXECUTED.  
 02AB 789 ALL REGISTERS EXCEPT R0-R4 ARE PRESERVED.  
 02AB 790  
 02AB 791 --  
 02AB 792  
 02AB 793 DL\_STARTIO: :START I/O OPERATION  
 02AB 794  
 02AB 795  
 02AB 796 COMPUTE PHYSICAL MEDIA ADDRESS  
 02AB 797  
 02AB 798 LBN = LBN \* (SECTORS/BLOCK)  
 02AB 799 LBN/(SECTORS/TRACK) = D + SECTOR  
 02AB 800 D/(TRACKS/CYLINDER) = CYLINDER + TRACK  
 02AB 801  
 02AB 802  
 02AB 803  
 02AB 804 PREPROCESS UCB FIELDS  
 02AB 805  
 02AB 806  
 02AB 807 PREPROCESS:  
 38 A3 DD 02AB 808 MOVL IRPSL\_MEDIA(R3),- : Copy given MEDIA address (logical)  
 00BC C5 02AB 809 UCBSL\_MEDIA(R5),- : to the UCB.  
 0B E0 02AE 810 BBS #IRPSL\_PHYSIO,- : IF SET - PHYSICAL I/O

02AB 808  
 02AB 809  
 02AE 810

50 00BC C5 26 2A A3	02	C5 02B0 011	MULL3 #2_UCBSL_MEDIÁ(R5), R0	:SCALE LBN IN R0
52 44 A5		C5 02B3 012	MOVZBL UCBSB_SECTORS(R5),R2	:GET NUMBER OF SECTORS PER TRACK
00BC C5 50 50 52		9A 02B9 013	CLRL R1	:CLEAR HIGH PART OF DIVIDEND
52 45 A5		D4 02BD 014	EDIV R2, R0, R0, UCBSL MEDIA(R5)	:CALCULATE SECTOR NUMBER AND STORE
51 50 50 52		7B 02BF 015	MOVZBL UCBSB_TRACKS(R5),R2	:GET NUMBER OF TRACKS PER CYLINDER
00BD C5 51		9A 02C6 016	EDIV R2, R0, R0, R1	:CALCULATE TRACK AND CYLINDER
00BE C5 50		7B 02CA 017	MOVB R1,UCBSL_MEDIA+1(R5)	:STORE TRACK NUMBER
		90 02CF 018	MOVW R0,UCBSL_MEDIA+2(R5)	:STORE CYLINDER NUMBER
		80 02D4 019		
		02D9 020	108:	
		0081 C5 90	MOVB UCBSB_ERTMAX(R5),-	:INITIALIZE ERROR RETRY COUNT
00C0 C5 7E A5		02D9 021	UCBSB_ERTCNT(R5)	
00D6 C5 B4		02DD 022	MNEGW UCBSW_BCNTR5, UCBSW_BCR(R5)	:INIT NEG BYTES LEFT TO XFER
		02E0 023	CLRW UCBSW_DL_DPN(R5)	:CLEAR DATA PATH NO. FOR USE AS- UBA RESOURCE ALLOCATION FLAG
009A C5 00E4 C5		02EA 024	CLRB UCB\$B_DL_DPPF(R5)	:CLEAR DATAPATH PURGE ERROR REGISTER
20 A3 94		02EE 025	MOVW IRPSW_FUNC(R3),UCBSW_FUNC(R5)	:SAVE FUNCTION CODE
51 20 A3 00		02F4 026	EXTZV #IRPSV_FCODE,-	:EXTRACT I/O FUNCTION CODE
0092 C5 51		02F6 027	MOVW #IRPSV_FCODE,IRPSW_FUNC(R3),R1	
51 02 91		02FA 028	CMPB R1,UCBSB_FEX(R5)	:STORE FUNCTION DISPATCH INDEX
06 12		0302 029	BNEQ #IOS_SEEK,R1	:SEEK FUNCTION?
38 A3 80		0304 030	MOVW 208	:IF NEQ - NO
00BE C5		0307 031	IRPSL_MEDIA(R3),-	:STORE CYLINDER ADDRESS
		030A 032	UCBSW_DC(R5)	:...
		030A 033		
		030A 034		
		030A 035	208:	
		02 AA 030A	BICW #UCBSM_DIAGBUF,-	
68 A5 02		030C 036	UCBSW_DEVSTS(R5)	:CLR DIAGNOSTIC BUFFER PRESENT
07 E1		030E 037	BBC #IRPSV_DIAGBUF,-	:IF CLR - NO DIAG BUFFER
04 2A A3		0310 038	IRPSW_STS(R3),FDISPATCH	
68 A5 02		0313 039	BISW #UCBSM_DIAGBUF,UCBSW_DEVSTS(R5)	:SET DIAG BUFFER PRESENT
		0317 040		
		0317 041		
		0317 042		
		0317 043		
		0317 044		
		0317 045		
		0317 046		
		0317 047		
		0317 048		
		0318 049		
		031D 050		
		0320 051		
		0322 052		
		0325 053		
		032A 054		
		032D 055		
		0331 056		
		0336 057		
		0336 058		
		0336 059		
		0336 060		
		0336 061		
		0336 062		
		0336 063		
		0336 064		
		0336 065		
		0336 066		
		0336 067		
			108:	
53 58 A5 D0		0317 046	FDISPATCH:	
08 E0		0317 047	MOVL UCB\$L_IRP(R5),R3	:FUNCTION DISPATCH
0D 2A A3 E0		0318 048	BBS #IRPSV_PHYSIO,-	:GET IRP ADDRESS
08 E0		031D 049	IRPSW_STS(R3),10\$	:IF SET - PHYSICAL I/O FUNCTION
50 08 64 A5 0SAE		0320 050	BBS #UCBSV_VALID,-	:IF SET - VOLUME SOFTWARE VALID
0254 BF SC		0322 051	UCBSW_STS(R5),10\$	
00C9 C5 94		0325 052	MOVZWL \$SSS_VOLINV,R0	:SET VOLUME INVALID STATUS
0092 C5 9A		032A 053	BRW RESETXFR	:RESET BYTE COUNT AND EXIT
		032D 054	CLRB UCB\$B_DL_DCHEK(R5)	:CLEAR DATA CHECK IN PROGRESS
		0331 055	MOVZBL UCBSB_FEX(R5),R3	:GET FUNCTION DISPATCH INDEX
		0336 056	CASE R3,<-	:DISPATCH TO FUNCTION HANDLING ROUTINE
		0336 057	UNLOAD,-	UNLOAD
		0336 058	SEEK,-	SEEK
		0336 059	NOP,-	RECALIBRATE (unsupported)
		0336 060	DRVCLR,-	DRVCLR
		0336 061	NOP,-	RELEASE PORT (unsupported)
		0336 062	NOP,-	OFFSET HEADS (unsupported)
		0336 063	NOP,-	RETURN TO CENTER (unsupported)
		0336 064	PACKACK,-	PACK ACKNOWLEDGE
		0336 065	NOP,-	SEARCH (unsupported)
		0336 066	WRITECHECK,-	WRITE CHECK
		0336 067	WRITEDATA,-	WRITE DATA

		0336	868	READDATA,-	: READ DATA
		0336	869	NOP -	: WRITE HEADER (unsupported)
		0336	870	READHEAD,-	: READ HEADER
		0336	871	NOP,-	: place holder
		0336	872	NOP,-	: place holder
		0336	873	AVAILABLE=	: AVAILABLE
		0336	874	>,LIMIT=#CDF_UNLOAD	
		035C	875		
		035C	876	NOP:	: NO-OP
		035C	877	SEEK:	: SEEK
		035C	878	DRVCLR:	: DRIVE CLEAR (GET STATUS & RESET)
		035C	879	DO_FUNCTION:	
	27 11	0360	880	EXFUNCL RETRYERR	: EXECUTE FUNCTION - RETRY IF FAILURE
		0362	881	BRB NORMAL	: SUCCESSFUL - EXIT WITH NORMAL STATUS
64 AS 0800 8F A8		0362	883	PACKACK:	
		0368	884	BISW #UCBSM_VALID,-	: PACK ACKNOWLEDGE (GET STATUS & RESET)
	F2 11	0368	885	UCBSW_STS(R5)	: Set software volume valid bit.
		036A	886	BRB DO_FUNCTION	: Then go do hardware function.
		036A	887		
64 AS 0800 8F AA		036A	888	UNLOAD:	: UNLOAD
		0370	889	AVAILABLE:	: AVAILABLE
	17 11	0370	890	BICW #UCBSM_VALID,-	: Clear software volume valid bit.
		0372	891	UCBSW_STS(R5)	: and go complete operation without
		0372	892	BRB NORMAC	: any hardware interaction.
4000 8F AA 009A C5		0372	894	WRITECHECK:	: WRITE CHECK
		0376	895	READHEAD:	: READ HEADER
		0379	896	BICW #IOSM_DATACHECK,-	: CLEAR DATA CHECK REQUEST-
		0379	897	UCBSW_FUNC(R5)	: TO PREVENT EXTRA WRITE CHECK
		0379	898		
		0379	899	WRITEDATA:	: WRITE DATA
		0379	900	READDATA:	: READ DATA
		0380	901	EXFUNCL RETRYERR,F_SEEK	: EXECUTE EXPLICIT SEEK - RETRY IF FAIL
53 0092 C5 9A		0380	902		
		0380	903	MOVZBL UCBSB_FEX(R5),R3	: GET FUNCTION DISPATCH INDEX
		0385	904	EXFUNCL RETRYERR	: EXECUTE TRANSFER FUNCTION
		0389	905	:	
		0389	906	:	
		0389	907	OPERATON COMPLETION	
		0389	908	:	
		0389	909		
50 01 0057 3C		0389	910	NORMAL:	: SUCCESSFUL OPERATION COMPLETE
		0389	911	MOVZWL #SSS_NORMAL,RO	: SET NORMAL COMPLETION STATUS
		038C	912	BRW FUNCXT	: FUNCTION EXIT
		038F	913		
0080 C5 97 03		038F	914	RETRYERR:	: RETRIABLE ERROR
		038F	915	DEC B	: ANY RETRIES LEFT?
		0393	916	BEQL FATALERR	: IF EQL - NO
	FF7F 31	0395	917	BRW FDISPATCH	: RETRY FUNCTION
		0398	918		
50 0254 8F 09		0398	919	FATALERR:	: UNRECOVERABLE ERROR
		0390	920	MOVZWL #SSS_VOLINV,RO	: ASSUME VOLUME INVALID STATUS
43 00D4 C5		039F	921	BBS #RL_AP_V_VC -	: IF SET - VOLUME INVALID
		03A3	922	UCBSW_DL_MP(R5),FUNCXT	: ...
50 025C 8F 3C		03A3	923		
		03A3	924	MOVZWL #SSS_WRTLCK,RO	: ASSUME WRITE LOCK ERROR STATUS

06 00D4	0D	E1	03A8	925	BBC	#RL_MP_V_WL - UCBSW_DL_MP(R5),58	:IF CLR - VOLUME NOT WRITE LOCKED
32 00D4	C5	EO	03AA	926	BBS	#RL_MP_V_WGE - UCBSW_DL_MP(R5),FUNCXT	:IF SET - WRITE GATE ERROR :IF WL & WGE SET - WRITE LOCK ERROR
50 005C	8F	3C	03B4	930	58:	MOVZWL #SSS_DATACHECK, R0	:ASSUME DATA CHECK ERROR STATUS
00C9	C5	95	03B9	931	TSTB	UCBSB_DL_DCHEK(R5)	:WRITE CHECK IN PROGRESS?
OC	13	03BD	932		BEQL	108	:IF EQL - NO
06 00CE	C5	EO	03BF	933	BBS	#RL_CS_V_OPI - UCBSW_DL_CS(R5),108	:IF SET - NOT WRITE CHECK ERROR
OB	0B	EO	03C5	935	BBS	#RL_CS_V_CRC - UCBSW_DL_CS(R5),FUNCXT	:IF SET - WRITE CHECK ERROR
1B 00CE	C5	EO	03C7	936			...
50 01F4	8F	3C	03C8	937	MOVZWL	#SSS_PARITY, R0	:ASSUME PARITY ERROR STATUS
OB	0B	EO	03D0	938	BBS	#RL_CS_V_CRC - UCBSW_DL_CS(R5),FUNCXT	:IF SET - (CRC ERROR :OR DATAPATH PURGE ERROR)
10 00CE	C5	EO	03D2	940			
50 008C	8F	3C	03D6	941	MOVZWL	#SSS_DRVERR, R0	:ASSUME DRIVE ERROR STATUS
OE	0E	EO	03DB	942	BBS	#RL_CS_V_DE - UCBSW_DL_CS(R5),FUNCXT	:IF SET - DRIVE ERROR
05 00CE	C5	EO	03DD	944			...
50 0054	8F	3C	03E1	945	MOVZWL	#SSS_CTRLERR, R0	:ASSUME CONTROLLER ERROR STATUS
			03E6	946			
			03E6	947			
			03E6	948	FUNCXT:	PUSHL R0	:FUNCTION EXIT
00000000	GF	DD	03E6	949		G^IOC\$DIAGBUFILL	:SAVE FINAL REQUEST STATUS
0092	C5	16	03E8	950		#CDF_WRITECHECK,UCBSB_FEX(R5)	:FILL DIAGNOSTIC BUFFER IF PRESENT
0A	91	03EE	951			;DRIVE RELATED FUNCTION?	
2D	1A	03F3	952			:IF GTRU - YES	
0092	C5	11	03F5	953		#CDF_AVAILABLE,UCBSB_FEX(R5)	:DRIVE RELATED FUNCTION?
26	13	03FA	954			:IF EQL - YES	
53 58	A5	00	03FC	955		UCBSL_IRP(R5),R3	:RETRIEVE ADDRESS OF IRP
00C0	C5	A1	0400	956		UCBSW_BCR(R5),-	:CALCULATE BYTES TRANSFERRED
02 AE	32	A3	0404	957		IRPSW_BCNT(R3),2(SP)	
00D6	C5	B5	0408	958		UCBSU_DL_DPN(R5)	:ARE UBA RESOURCES ALLOCATED?
1A	13	040C	959			:IF EQL - NO	
01	E1	040E	960			:ADAPTER MAPPING?	
OE 00F6	C5	0410	961			:IF BC NO	
		0414	962			:RELEASE DATA PATH	
		041A	963			:RELEASE MAP REGISTERS	
06	11	0420	964	108:		208	:JOIN COMMON CODE
00D8	C5	00	0422	965		MOVL UCBSL_DL_SVAPTE(R5),-	:RESTORE ORIGINAL SVAPTE
78 A5	0426	966				UCBSL_SVAPTE(R5)	
		0428	967	208:			:RELEASE CHANNEL IF OWNED
S1 D4	042E	968					
50 8ED0	0430	969					
		0433	970				
			0433	971			
					CLRL R1		:CLEAR SECOND STATUS LONGWORD
					POPL R0		:RETRIEVE FINAL REQUEST STATUS
					REQCOM		:COMPLETE REQUEST

0439 973 : FEXL - RL11 HARDWARE FUNCTION EXECUTION  
0439 974 :  
0439 975 : THIS ROUTINE IS CALLED VIA A BSB WITH A BYTE IMMEDIATELY FOLLOWING THAT  
0439 976 : SPECIFIES THE ADDRESS OF AN ERROR ROUTINE. ALL DATA IS ASSUMED TO HAVE BEEN  
0439 977 : SET UP IN THE UCB BEFORE THE CALL. THE APPROPRIATE PARAMETERS ARE LOADED  
0439 978 : INTO DEVICE REGISTERS AND THE FUNCTION IS INITIATED. THE RETURN ADDRESS  
0439 979 : IS STORED IN THE UCB AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE  
0439 980 : INTERRUPT OCCURS, CONTROL IS RETURNED TO THE CALLER.  
0439 981 :  
0439 982 :  
0439 983 :  
0439 984 :  
0439 985 : R3 = FUNCTION TABLE DISPATCH INDEX  
0439 986 : R5 = DEVICE UNIT UCB ADDRESS  
0439 987 :  
0439 988 : 00(SP) = RETURN ADDRESS OF CALLER  
0439 989 : 04(SP) = RETURN ADDRESS OF CALLER'S CALLER  
0439 990 :  
0439 991 : IMMEDIATELY FOLLOWING INLINE AT THE CALL SITE IS A BYTE WHICH CONTAINS  
0439 992 : A BRANCH DESTINATION TO AN ERROR RETRY ROUTINE.  
0439 993 :  
0439 994 :  
0439 995 :  
0439 996 :  
0439 997 :  
0439 998 :  
0439 999 :  
0439 1000 :  
0439 1001 :  
0439 1002 :  
0439 1003 :  
0439 1004 :  
0439 1005 :  
0439 1006 :  
0439 1007 :  
0439 1008 :  
0439 1009 :  
0439 1010 :  
0439 1011 :  
0439 1012 :  
0439 1013 :  
0439 1014 :  
0439 1015 :  
0439 1016 :  
0439 1017 :  
0439 1018 :  
0439 1019 :  
INPUTS:  
R3 = FUNCTION TABLE DISPATCH INDEX  
R5 = DEVICE UNIT UCB ADDRESS  
00(SP) = RETURN ADDRESS OF CALLER  
04(SP) = RETURN ADDRESS OF CALLER'S CALLER  
IMMEDIATELY FOLLOWING INLINE AT THE CALL SITE IS A BYTE WHICH CONTAINS  
A BRANCH DESTINATION TO AN ERROR RETRY ROUTINE.  
OUTPUTS:  
THERE ARE FOUR EXITS FROM THIS ROUTINE:  
1. SPECIAL CONDITION - THIS EXIT IS TAKEN IF A POWER FAILURE OCCURS  
OR THE OPERATION TIMES OUT. IT IS A JUMP TO THE APPROPRIATE  
ERROR ROUTINE.  
2. FATAL ERROR - THIS EXIT IS TAKEN IF A FATAL CONTROLLER OR DRIVE  
ERROR OCCURS OR IF ANY ERROR OCCURS AND ERROR RETRY IS EITHER  
INHIBITED OR EXHAUSTED. IT IS A JUMP TO THE FATAL ERROR EXIT  
ROUTINE.  
3. RETRIEABLE ERROR - THIS EXIT IS TAKEN IF A RETRIEABLE CONTROLLER  
OR DRIVE ERROR OCCURS AND ERROR RETRY IS NEITHER INHIBITED  
NOR EXHAUSTED. IT CONSISTS OF TAKING THE ERROR BRANCH EXIT  
SPECIFIED AT THE CALL SITE.  
4. SUCCESSFUL OPERATION - THIS EXIT IS TAKEN IF NO ERRORS OCCUR  
DURING THE OPERATION. IT CONSISTS OF A RETURN INLINE.  
IN ALL CASES IF AN ERROR OCCURS, AN ATTEMPT IS MADE TO LOG THE ERROR.  
IN ALL CASES FINAL DEVICE REGISTERS ARE RETURNED VIA THE UCB.  
UCBSW\_BCR(R5) = NEGATIVE BYTES REMAINING TO TRANSFER

0093	C5	BED0	0439	1021	FEXL:			
50	53	90	0439	1022		POPL	UCBSL_DPC(R5)	:FUNCTION EXECUTOR
51	24	A5	043E	1023		MOV B	R3,UCBSB_CEX(R5)	:SAVE DRIVER PC VALUE
04	2C	A0	0447	1024		MOVL	UCBSL_CRB(R5),R0	:SAVE CASE INDEX
						MOVL	CRBSL_INTD+VECSL_IDB(R0)	:GET ADDRESS OF PRIMARY CRB
						CMPL	R5, IDBSL_OWNER(RT)	,R1 :GET ADDRESS OF IDB
						BNEQ	10\$	:DOES THIS PROCESS OWN CHANNEL?
54	55	01	044B	1025		MOVL	IDBSL_CSR(R1),R4	:IF NEQ - NO
	05	12	044F	1026		BRB	20\$	:SET ASSIGNED CHANNEL CSR ADDRESS
	61	00	0451	1027				
	06	11	0454	1028				
			0456	1029				
			0456	1030	10\$: REQPCHAN			:REQUEST CHANNEL (RETURNS R4 = CSR ADR)
			045C	1031				
			045C	1032	20\$: CASE	R3,<-		:DISPATCH TO PROPER FUNCTION ROUTINE
			045C	1033		IMMED,-		:NO OPERATION
			045C	1034		IMMED,-		:UNLOAD VOLUME (NOP)
			045C	1035		POSIT,-		:SEEK CYLINDER
			045C	1036		IMMED,-		:RECALIBRATE (NOP)
			045C	1037		DRCLR,-		:DRIVE CLEAR (GET STATUS & RESET)
			045C	1038		IMMED,-		:RELEASE DRIVE (NOP)
			045C	1039		IMMED,-		:OFFSET HEADS (NOP)
			045C	1040		IMMED,-		:RETURN TO CENTERLINE (NOP)
			045C	1041		DRCLR,-		:PACK ACKNOWLEDGE
			045C	1042		IMMED,-		:SEARCH (NOP)
			045C	1043		>		
01A3	31	0474	1044			BRW	XFER	:TRANSFER FUNCTION

0477 1046	; IMMEDIATE FUNCTION EXECUTION		
0477 1047			
0477 1048	FUNCTIONS INCLUDE:		
0477 1049			
0477 1050			
0477 1051	NO OPERATION,		
0477 1052	DRIVE CLEAR, AND		
0477 1053	PACK ACKNOWLEDGE		
0477 1054			
0477 1055	INPUTS:		
0477 1056	R3	- CASE INDEX	
0477 1057	R4	- CSR ADDRESS	
0477 1058	R5	- UCB ADDRESS	
0477 1059			
0477 1060	FUNCTIONAL DESCRIPTION:		
0477 1061			
0477 1062	INTERRUPTS ARE LOCKED OUT. THE APPROPRIATE FUNCTION IS INITIATED WITH		
0477 1063	INTERRUPT ENABLE, AND A WAITFOR INTERRUPT AND KEEP CHANNEL IS EXECUTED.		
0477 1064			
0477 1065			
0477 1066	DRCLR:		
04 A6 08 A8	0477 1067	BISW	#RL_DA_M_STS!- RL_DA_R_RST!RL_DA_M_MRK,RL_DA(R4) ;...
	0478 1068		: DRIVE CLEAR
	0478 1069		: SET GETSTATUS,RESET,AND MARK IN DAR
	0478 1070	IMMED:	
	0478 1071	CKPWR	: IMMEDIATE FUNCTION EXECUTION
64 FB9E CF43 52 A9	0494 1072		: DISABLE INTERRUPTS, CHECK POWER,-
	0498 1073	BISW3 R2,FTAB[R3],RL_CS(R4)	: AND PUT UNIT NUMBER IN R2<9:8>
	0498 1074	WF KPCH RETREG,#2	: MERGE UNIT WITH FNTN AND EXECUTE
	04A5 1075	IOFORK	: WAITFOR INTERRUPT
	04AB 1076		: RETURN FROM ISR-
0371 31 04AB	1077	BRW RETREG	: CREATE FORK PROCESS (GJSB BACK TO ISR)

04AE 1079 :  
 04AE 1080 : POSITIONING FUNCTION EXECUTION  
 04AE 1081 :  
 04AE 1082 : FUNCTIONS INCLUDE:  
 04AE 1083 :  
 04AE 1084 : SEEK CYLINDER  
 04AE 1085 :  
 04AE 1086 :  
 04AE 1087 : INPUTS:  
 R3 - CASE INDEX  
 04AE 1088 : R4 - DEVICE CSR ADDRESS  
 04AE 1089 : R5 - UCB ADDRESS  
 04AE 1090 :  
 04AE 1091 :  
 04AE 1092 :  
 04AE 1093 : FUNCTIONAL DESCRIPTION:  
 THE CYLINDER DIFFERENCE WORD IS CALCULATED AND LOADED INTO THE DISK  
 ADDRESS REGISTER, INTERRUPTS ARE LOCKED OUT, AND THE SEEK FUNCTION  
 IS INITIATED WITHOUT INTERRUPT ENABLE. THE CONTROLLER IS THEN POLLED  
 FOR READY, AND DEVICE INTERRUPTS ARE ENABLED.  
 04AE 1094 :  
 04AE 1095 :  
 04AE 1096 :  
 04AE 1097 :  
 04AE 1098 : SINCE THE RL01/RL02 DO NOT ISSUE AN INTERRUPT UPON COMPLETION OF A  
 04AE 1099 : SEEK, OVERLAPPED SEEKS ARE NOT ATTEMPTED, AND ONE OF THE FOLLOWING IS  
 04AE 1100 : PERFORMED.  
 04AE 1101 :  
 04AE 1102 : IF ONLY A SEEK FUNCTION IS BEING REQUESTED, A DUMMY READ HEADER  
 04AE 1103 : FUNCTION IS ISSUED AND A WAITFOR INTERRUPT IS INITIATED.  
 04AE 1104 : THE READ HEADER IS USED TO SIGNAL THE END OF THE SEEK, SINCE IT  
 04AE 1105 : WILL ISSUE AN INTERRUPT SHORTLY (315 USEC AVG) AFTER THE SEEK IS  
 04AE 1106 : COMPLETE. IT WILL ALSO SENSE FOR A TIMEOUT DURING THE SEEK.  
 04AE 1107 :  
 04AE 1108 :  
 04AE 1109 :  
 04AE 1110 :  
 04AE 1111 :  
 04AE 1112 :  
 04AE 1113 :  
 04AE 1114 :  
 04AE 1115 : POSIT: : POSITIONING FUNCTION  
 04AE 1116 :  
 04AE 1117 : OBTAIN CURRENT DISK ADDRESS  
 04AE 1118 :  
 04AE 1119 : IF THERE HAS NOT BEEN A PREVIOUS TRANSFER DURING THIS REQUEST,  
 04AE 1120 : A READ HEADER IS EXECUTED TO DETERMINE THE CURRENT DISK ADDRESS.  
 04AE 1121 :  
 00D6 C5 B5 04AE 1122 TSTW UCBSW\_DL\_DPN(R5) : WAS THERE A PREVIOUS TRANSFER?  
 09 13 04B2 1123 BEQL 108 : IF EQL - NO, READ HEADER  
 S1 00D2 C5 3F AB 04B4 1124 BICW3 #^077,UCBSW\_DL\_DA(R5),R1 : PUT CURRENT CYL & SURFACE IN R1  
 00B5 31 04BA 1125 BRW 608 : CALCULATE DIFFERENCE WORD  
 04BD 1126  
 S3 08 9A 04BD 1127 108: MOVZBL #8,R3 : SET READ HEADER RETRY COUNT IN R3  
 04C0 1128 208: CKPWR : DISABLE INTERRUPTS, CHECK POWER,-  
 04D9 1129 04D9 R2,EF READHEAD!RL\_CS\_M\_IE,- ;EXECUTE READ HEADER  
 64 04DE 1130 RL\_CSTR4  
 04DF 1131 WFICKPCH 408,02 : WAIT FOR INTERRUPT OR TIMEOUT  
 04E9 1132 IOFORK : CREATE FORK PROCESS  
 00CE C5 8000 8F B3 04EF 1133 BITW #RL\_CS\_M\_CE,UCBSW\_DL\_CS(R5) : ANY ERRORS?  
 74 13 04F6 1134 BEQL 508 : IF EQL - NO

53 97 04FB 1136  
 C4 12 04FA 1137  
 04FC 1138  
 04FC 1139  
 0081 8F 80 04FC 1140  
 04 A4 0500 1141  
 0502 1142  
 051B 1143  
 0046 BF 52 A9 051B 1144  
 64 0520 1145  
 0521 1146  
 0528 1147  
 0531 1148  
 0048 BF 52 A9 054A 1149  
 64 054F 1151  
 0550 1152  
 055A 1153  
 00CE CS 8000 8F 03 B3 0560 1154  
 13 0567 1155  
 0569 1156 40\$:  
 02B3 31 0569 1157 :  
 S1 00D4 CS 3F AB 056C 1158 50\$:  
 056C 1159 50\$:  
 0572 1160  
 0572 1161  
 0572 1162  
 0572 1163 : CALCULATE CYLINDER DIFFERENCE WORD  
 0572 1164 :  
 0572 1165 :  
 50 01 06 00BD CS 50 D4 0572 1166 60\$: CLPL R0  
 50 09 07 00BE CS 51 F0 0574 1167  
 51 50 B1 0582 1168  
 52 13 0585 1169  
 51 007F 8F AA 0587 1170  
 50 007F 8F AA 058C 1171  
 51 50 A2 0591 1172  
 08 13 0594 1173  
 06 1E 0596 1174  
 51 51 AE 0598 1175  
 51 04 AB 0598 1176  
 04 A4 51 01 A9 059E 1177 70\$: INSV UCBSW\_DA+1(R5),#6,#1,R0  
 05A5 1178 70\$: BISW3 #RL\_DA\_M\_MRK,R1,RL\_DA(R4) ;SET MARKER AND LOAD DIFFERENCE WORD  
 05AA 1179  
 05AA 1180  
 05AA 1181 : EXECUTE SEEK  
 05AA 1182 :  
 05AA 1183 :  
 05AA 1184 :  
 05AA 1185 : CKPWR  
 05C3 1186 :  
 0046 BF 52 A9 05C3 1187 BISW3 R2,WF SEEK!RL\_CS\_M\_IE,-  
 64 05C8 1188 RL\_CSTR4)  
 05C9 1189 WF IKPCH 40\$,#2  
 0092 CS 02 91 05D3 1190 IOFORK  
 08 13 05D9 1191 80\$: CMPB #10\$\_SEEK,UCBSB\_FEX(R5)  
 05DE 1192 BEQL 90\$ ;DISABLE INTERRUPTS, CHECK POWER,-  
 ;AND PUT UNIT NUMBER IN R2<9:8>  
 ;EXECUTE SEEK FUNCTION  
 ;WAIT FOR SEEK TO BEGIN (INTERRUPT)  
 ;CREATE FORK PROCESS  
 ;IS SEEK ASSOCIATED WITH A TRANSFER?  
 ;IF EQ - NO, SEEK ONLY

			OSE0	1193			
			OSE0	1194			
			OSE0	1195	:: RETURN FOR SEEK ASSOCIATED WITH A TRANSFER REQUEST		
			OSE0	1196			
			OSE0	1197			
009C	C5	D6	OSE0	1198	INCL UCBSL_DPC(R5)		:ADJUST TO CORRECT RETURN ADDRESS
009C	D5	17	OSE4	1199	JMP <del>UCBSC_DPC(R5)</del>		:RETURN TO DRIVER FOR TRANSFER
			OSE8	1200			
			OSE8	1201	:: RETURN FOR SEEK ONLY REQUEST		
			OSE8	1202			
			OSE8	1203			
			OSE8	1204	90\$: CKPWR		:DISABLE INTERRUPTS, CHECK POWER,-
			OSE01	1205			:AND PUT UNIT NUMBER IN R2<9:8>
0048	8F	A9	OSE01	1206	BISW3 R2,#F READHEAD!RL_CS_M_I&,-		:EXECUTE DUMMY READ HEADER
			OSE06	1207	RL_CSTR4)		
			OSE07	1208	WF IKPCH RETREG,#2		
			OSE11	1209	IOFORK		:WAIT FOR SEEK TO COMPLETE (INTERRUPT)
0205	31	0617	1210		BRW RETREG		:CREATE FORK PROCESS

061A 1212

061A 1213

061A 1214

TRANSFER FUNCTION EXECUTION

061A 1215

061A 1216

061A 1217

061A 1218

061A 1219

061A 1220

061A 1221

061A 1222

061A 1223

061A 1224

061A 1225

061A 1226

061A 1227

061A 1228

061A 1229

061A 1230

061A 1231

061A 1232

061A 1233

061A 1234

061A 1235

061A 1236

061A 1237

061A 1238

061A 1239

061A 1240

061A 1241

061A 1242

061A 1243 XFER:

TRANSFER FUNCTIONS INCLUDE:  
WRITE CHECK  
WRITE DATA  
READ DATA, AND  
READ HEADER

INPUTS:

R3 - CASE INDEX  
R4 - DEVICE CSR ADDRESS  
R5 - UCB ADDRESS

FUNCTIONAL DESCRIPTION:

A UNIBUS DATAPATH IS REQUESTED FOLLOWED BY THE APPROPRIATE NUMBER OF MAP REGISTERS REQUIRED FOR THE TRANSFER. THE TRANSFER PARAMETERS ARE LOADED INTO THE DEVICE REGISTERS, INTERRUPTS ARE LOCKED OUT, THE FUNCTION IS INITIATED, AND A WAITFOR INTERRUPT AND KEEP CHANNEL IS EXECUTED.

UPON RETURN FROM THE INTERRUPT SERVICE ROUTINE, IF THE TRANSFER IS COMPLETE, THE APPROPRIATE EXIT IS TAKEN. IF THE FUNCTION IS NOT COMPLETE TRANSFER PARAMETERS ARE UPDATED AND A RETURN TO FDISPATCH IS EXECUTED TO RE-ISSUE SEEK AND TRANSFER FUNCTIONS WHILE KEEPING CHANNEL AND UBA RESOURCES. IF A DATA CHECK HAS BEEN REQUESTED, IT IS PERFORMED BEFORE RETURNING TO FDISPATCH.

00EC C5 3E 00F6	01	E0	061A 1244	BBS #UCBSV_DL_MAPPING,-	: TRANSFER FUNCTION EXECUTION
00F2	C5	00F	061C 1245	UCBSU_DL_FLAGS(R5),28	: ADAPTER MAPPING?
50 00F4	C5	80	0620 1246	MOVW UCBSA_DL_BUF_PA(R5)	: BRANCH IF ADAPTER MAPPING.
08 A4	50	3C	0627 1247	MOVZWL UCBSA_DL_BUF_PA+2(R5),R0	: GET 1ST WORD OF BUFFER ADDR
50 50	04	80	0630 1248	MOVW R0,RL_BAE(R4)	: GET BITS 16:21 OF BUFFER ADDRESS
00EB C5	50	78	0630 1249	ASHL #4,R0,R0	: SET MEMORY EXTENSION BITS IN BAE
		90	0634 1250	MOVB R0,UCBSB_DL_XBA(R5)	: PUT MEMORY EXTENSION BITS IN <5:4>
			0639 1251		: OF CSR

FIRST TRANSFER OF THIS I/O REQUEST - ALLOCATE RESOURCES

00D6 C5 00D6	85	0639	1254	TSTW UCBSU_DL_DPN(R5)	: RESOURCES ALREADY ALLOCATED?
61	12	063D	1255	BNEQ \$8	: IF NEQ - YES
00E0 C5 00E0	D4	063F	1256	CLRL UCBSA_DL_MVRTRN(R5)	: ASSUME READ
53	08	91	0643	CMPB #CDF_WRITEDATA,R3	: WRITE DATA?
09	12	0646	1258	BNEQ 18	: IF NEQ NO
00000000 GF	9E	0648	1259	MOVAB G^IOCSMOVFRUSER,-	: SET MOVE ROUTINE ADDRESS FOR
00E0 C5	00E0	064E	1260	UCBSA_DL_MVRTRN(R5)	: 1ST PARTIAL WRITE
00D8 C5 00D8	78 A5	0651	1261 18:	MOVL UCBSL_SVAPTE(R5),UCBSL_DL_SVAPTE(R5)	: SAVE SVAPTE FOR BUFFER COPY
00D6 C5 00D6	01 AE	0657	1262	MNEGW #1,UCBSU_DL_DPN(R5)	: SET FIRST XFER FLAG
	42 11	065C	1263	BRB \$8	: JOIN COMMON CODE
		065E	1264		
		065E	1265		
		065E	1266		
		065E	1267		

FIRST TRANSFER OF THIS I/O REQUEST - ALLOCATE RESOURCES

00D6 C5 00D6	85	065E	1268 28:	TSTW UCBSU_DL_DPN(R5)	: UBA RESOURCES ALREADY ALLOCATED?
--------------	----	------	----------	-----------------------	------------------------------------

3C 12 0662 1269  
 0664 1270  
 066A 1271  
 0670 1272  
 0676 1273  
 067A 1274  
 067D 1275  
 0680 1276  
 0685 1277  
 0689 1278  
 068C 1279  
 068F 1280  
 0694 1281  
 069A 1282  
 06A0 1283  
 06A0 1284  
 06A0 1285  
 06A0 1286  
 06A0 1287  
 06A0 1288  
 06A0 1289  
 06A0 1290  
 06A0 1291  
 06A0 1292  
 06A0 1293  
 06A0 1294  
 06A3 1295  
 06A3 1296  
 06A3 1297  
 06A3 1298  
 06A3 1299  
 06A9 1300  
 06AD 1301  
 06B0 1302  
 06B4 1303  
 06B9 1304  
 06BC 1305  
 06C1 1306  
 06C6 1307  
 06C8 1308  
 06CD 1309  
 06CD 1310  
 06CD 1311  
 06CD 1312  
 06DD 1313  
 06DD 1314  
 06E0 1315  
 06D5 1316  
 06DB 1317  
 06E1 1318  
 06E5 1319  
 06E5 1320  
 06EA 1321  
 06F1 1322  
 06F8 1323  
 06FC 1324  
 06FC 1325 :

BNEQ 58 : IF NEQ - YES  
 REQDPR : REQUEST DATAPATH  
 REQMPR : REQUEST MAP REGISTERS  
 LOADUBA : LOAD UNIBUS MAP REGISTERS  
 MOVL UCBSL\_CRB(R5),R1 : GET CRB ADDRESS  
 EXTZV #VEC\$V\_DATAPATH, #VEC\$S\_DATAPATH - : EXTRACT DATAPATH NUMBER -  
 CRBSL\_INTD+VECSB\_DATAPATH(R1),R0 : FOR UBA RESOURCE FLAG  
 MOVW R0,UCBSW\_DL\_DPN(R5) : INDICATE UBA RESOURCES ALLOCATED  
 MOVZWL UCBSW\_BOFF(R5),R0 : GET BYTE OFFSET IN PAGE  
 INSV CRBSL\_INTD+VECSW\_MAPREG(R1),- : INSERT HIGH 7 BITS OF ADDRESS  
 #9,#7,R0  
 MOVW R0,UCBSW\_DL\_SBA(R5) : SET BUFFER ADDRESS  
 EXTZV #7 #2 CRBSL\_INTD+VECSW\_MAPREG(R1),R0 : GET MEMORY EXTENSION BITS  
 MULB3 #16,R0,UCBSB\_DL\_XBA(R5) : POSITION MEMORY EXTENSION BITS TO <5:4>  
  
 :: COMMON TRANSFER POINT  
  
 FOR A READ OPERATION WHEN NO ADAPTER MAPPING IS PRESENT EMPTY THE  
 INTERNAL PHYSICALLY CONTIGUOUS BUFFER FROM THE PREVIOUS READ TO THE  
 USER'S BUFFER.  
  
 02DA 30 06A0 1293 :  
 5S: BSBW DL\_MOVE\_TO\_BUFFER ; COPY TO USER BUFFER  
 06A3 1294 :  
 06A3 1295 : PUT BUFFER ADDRESS, WORD COUNT, AND DISK ADDRESS IN DEVICE REGISTERS  
 06A3 1296 :  
 06A3 1297 :  
 06A3 1298 :  
  
 02 A4 00EC C5 80 06A3 1299 :  
 00C0 C5 AE 06A9 1300 :  
 00CC C5 06AD 1301 :  
 52 44 A5 9A 06B0 1302 :  
 S1 008C C5 9A 06B4 1303 :  
 S2 51 A2 06B9 1304 :  
 S2 0100 8F A4 06BC 1305 :  
 S2 00CC C5 81 06C1 1306 :  
 00CC C5 05 18 06C6 1307 :  
 00CC C5 52 80 06C8 1308 :  
 06CD 1309 :  
 06CD 1310 :  
 06CD 1311 :  
 06CD 1312 :  
 06DD 1313 :  
 06DD 1314 :  
 06E0 1315 :  
 06D5 1316 :  
 06DB 1317 :  
 06E1 1318 :  
 06E5 1319 :  
 06E5 1320 :  
 06EA 1321 :  
 06F1 1322 :  
 06F8 1323 :  
 06FC 1324 :  
 06FC 1325 :  
  
 MOVW UCBSW\_DL\_SBA(R5),RL\_BA(R4) : SET BUFFER ADDRESS  
 MNEGW UCBSW\_BCR(R5),- : GET BYTES LEFT TO TRANSFER AND -  
 UCBSW\_DL\_PBCR(R5) : ASSUME ONLY ONE TRANSFER NEEDED  
 UCBSB\_SECTORS(R5),R2 : GET SECTORS/SURFACE  
 MOVZBL UCBSW\_DA(R5),R1 : GET DESIRED SECTOR  
 SUBW R1 R2 : CALCULATE SECTORS LEFT ON SURFACE  
 MULW #256,R2 : CONVERT TO BYTES LEFT ON SURFACE  
 CMPW UCBSW\_DL\_PBCR(R5),R2 : ARE ADDITIONAL TRANSFERS REQUIRED?  
 BLEQU 10S : IF LEQU - NO  
 MOVW R2,UCBSW\_DL\_PBCR(R5) : SET BYTE COUNT FOR THIS TRANSFER  
  
 FOR A WRITE OPERATION WHEN NO ADAPTER MAPPING IS PRESENT  
 FILL INTERNAL PHYSICALLY CONTIGUOUS BUFFER FROM THE USER'S BUFFER.  
  
 02F2 30 06CD 1313 :  
 10S: BSBW DL\_MOVE\_FROM\_BUFFER ; COPY FROM USER BUFFER  
 06DD 1314 :  
  
 50 00EB C5 9A 06DD 1315 :  
 S2 50 F95E CF43 A8 06D5 1316 :  
 00CC C5 02 A7 06DB 1317 :  
 06 A4 52 AE 06E1 1318 :  
  
 51 008C C5 9A 06E5 1320 :  
 S1 06 008D C5 FO 06EA 1321 :  
 S1 09 07 008E C5 FO 06F1 1322 :  
 04 A4 51 BO 06F8 1323 :  
 06FC 1324 :  
 06FC 1325 :  
  
 MOVZBL UCBSB\_DL\_XBA(R5),R0 : SET MEMORY EXTENSION BITS  
 BISW FTAB[R3],R0 : MERGE XBA BITS WITH FUNCTION  
 DIVW3 #2,UCBSW\_DL\_PBCR(R5),R2 : CALCULATE TRANSFER WORD COUNT  
 MNEGW R2,RL\_MPTR4 : SET TRANSFER WORD COUNT  
  
 MOVZBL UCBSW\_DA(R5),R1 : PUT DESIRED SECTOR IN R1<5:0>  
 INSV UCBSW\_DA+1(R5),#6,#1,R1 : INSERT DESIRED SURFACE IN R1<6>  
 INSV UCBSW\_DC(R5),#7,#9,R1 : INSERT DESIRED CYLINDER IN R1<15:7>  
 MOVW R1,RL\_DA(R4) : SET DESIRED DISK ADDRESS

64 50 S2 A9 06FC 1326 : EXECUTE THE TRANSFER FUNCTION

06FC 1327 :  
 06FC 1328 :  
 06FC 1329 : CKPWR  
 0715 1330 :  
 0715 1331 :  
 0719 1332 : BISW3 R2, R0, RL CS(R4)  
 0723 1333 : WFIKPCH REFREG,#8  
 0723 1334 :  
 0729 1335 : IOFORK  
 0729 1336 :  
 0729 1337 :  
 0729 1338 : PURGE DATAPATH  
 0729 1339 :  
 0729 1340 :  
 00E4 C5 94 0729 1341 CLRBL UCBSB\_DL\_DPPE(R5) :CLEAR DATAPATH PURGE ERROR  
 00000000 GF 16 0720 1342 JSB G^1OC\$PURGDATA :PURGE DATAPATH  
 04 S0 E8 0733 1343 BLBS R0 20\$ :IF SET - NO PURGE ERRORS  
 00E4 C5 96 0736 1344 INCBL UCBSB\_DL\_DPPE(R5) :SET DATAPATH PURGE ERROR

073A 1345 :  
 073A 1346 : SAVE UBA REGISTERS FOR UPDATE AND REGDUMP ROUTINES  
 073A 1347 :  
 073A 1348 :  
 073A 1349 :  
 50 3E 00F6 C5 01 E1 073A 1350 20\$: BBC #UCBSV\_DL\_MAPPING,- :ADAPTER MAPPING?  
 50 00D8 C5 S1 D0 0740 1351 MOVL UCBSW\_DL\_FLAGS(R5),30\$ :IF BC NO  
 51 00CE C5 07 09 EF 0745 1352 EXTZV #9,#7,UCBSW\_DL\_BA(R5),R0 :SAVE DATAPATH REGISTER  
 50 02 07 04 EF 074C 1353 EXTZV #4,#2,UCBSW\_DL\_CS(R5),R1 :EXTRACT LOW BITS OF FINAL MAP REG NO.  
 50 02 07 S1 F0 0753 1354 INSV R1 #7,#2,R0 :EXTRACT HI BITS OF FINAL MAP REG NO.  
 50 01EF 8F B1 0758 1355 CMPW #495,R0 :INSERT HIGH BITS OF FINAL MAP REGISTER  
 50 05 18 075D 1356 BGEQ 258 :LEGAL MAP REGISTER NUMBER?  
 50 01EF 8F 3C 075F 1357 MOVZWL #495,R0 :IF GEO - YES  
 00DC C5 6240 D0 0764 1358 MOVL (R2)[R0],UCBSL\_DL\_FMPR(R5) :RESTRICT MAP REGISTER NUMBER  
 00E0 C5 D4 076A 1360 CLRL UCBSL\_DL\_PMPR(R5) :SAVE FINAL MAP REGISTER NUMBER  
 50 D7 076E 1361 DECL R0 :CLEAR PREVIOUS MAP REGISTER CONTENTS  
 0F 00 EC 0770 1362 CMPV #VEC\$V\_MAPREG,#VEC\$S\_MAPREG,- :CALCULATE PREVIOUS MAP REGISTER NUMBER  
 50 34 A3 0773 1363 CRBSL\_INTD+VEC\$U\_MAPREG(R3),R0 :...  
 06 14 0776 1364 BGTR 30\$: :IF GTR - NO  
 00E0 C5 6240 D0 0778 1365 MOVL (R2)[R0],UCBSL\_DL\_PMPR(R5) :SAVE PREVIOUS MAP REGISTER  
 03 00CE C5 OF E1 077E 1366 30\$: BBL R1 CS\_V\_CE,UCBSW\_DL\_CS(R5),40\$ :IF CLR - NO RL ERRORS  
 0098 31 0784 1367 BRW RETREG :DEVICE ERROR  
 03 00E4 C5 F9 0787 1368 40\$: BLBC UCBSB\_DL\_DPPE(R5),458 :IF CLR - NO PURGE ERROR  
 0090 31 078C 1369 BRW RETREG :PURGE ERROR

078F 1370 :  
 078F 1371 : RETURN HEADER INFORMATION FOR READ HEADER FUNCTION  
 078F 1372 :  
 078F 1373 :  
 078F 1374 :  
 0093 C5 0E 91 078F 1375 458: CMPB #CDF\_READHEAD,UCBSB\_CEX(R5) :READ HEADER FUNCTION?  
 2F 12 0794 1376 BNQ DATACHECK :IF NEQ - NO  
 00C0 C5 DD 0796 1377 PUSHL UCBSW\_BCR(R5) :SAVE NEG BYTES REMAINING  
 51 78 A5 DD 079A 1378 PUSHL UCBSL\_SVAPTE(R5) :SAVE ADDRESS OF PTE  
 52 06 DD 07A2 1379 MOVAB UCBSW\_DL\_DB(R5),R1 :SET ADDRESS OF INTERNAL BUFFER  
 7E A5 52 B1 07A5 1380 MOVL #6,R2 :SET NUMBER OF BYTES TO MOVE  
 06 1F 07A9 1382 CMPW R2 UCBSW\_BCNT(R5) :ROOM FOR FULL HEADER?  
 BLSSU 50\$ :IF LSSU - YES

00C0 C5 52 7E A5 3C 07AB 1383  
 52 7E A5 A3 07AF 1384 508: MOVZWL UCBSW\_BCNT(R5),R2 :SET LENGTH OF PARTIAL HEADER  
 00000000 GF 16 07B6 1385 SUBW3 UCBSW\_BCNT(R5),R2,UCBSW\_BCR(R5) :CALCULATE TRANSFER BYTE COUNT  
 78 A5 BED0 07BC 1386 JSB G^IOCSMOVTOUSER :MOVE HEADER TO USER BUFFER  
 00C0 C5 BED0 07C0 1387 POPL UCBSL\_SVAPTE(R5) :RESTORE ADDRESS OF PTE  
 07C5 1388 POPL UCBSU\_BCR(R5) :RESTORE NEG BYTES REMAINING

00C9 C5 00 009A DE E1 07C5 1390 : PERFORM DATA CHECK, IF REQUESTED  
 00 00 E4 07C7 1391 :  
 00C9 C5 96 07D1 1392 1393 DATACHECK:  
 53 0A 9A 07DS 1393 BBC #IOSV\_DATACHECK,- :DATACHECK AFTER PARTIAL TRANSFER  
 FE3F 31 07D8 1400 BBSC #0,UCBSB\_DL\_DCHEK(R5),- :IF CLR - DATA CHECK NOT REQUESTED  
 07DB 1401 UPDATE  
 07DB 1402 :  
 07DB 1403 : UPDATE BUFFER ADDRESS, CURRENT DISK ADDRESS, AND BYTES REMAINING  
 07DB 1404 : FOR NEXT TRANSFER  
 07DB 1405 :  
 07DB 1406 :  
 00CE C5 01 00F6 E1 07DB 1407 UPDATE: 1407 BBC #UCBSV\_DL\_MAPPING,- :UPDATE TRANSFER PARAMETERS  
 CF 8F 88 07DD 1408 UCBSW\_DL\_FLAGS(R5),103 :ADAPTER MAPPING?  
 00EB C5 00 00EB 07E1 1410 BICB3 #^XCF,UCBSW\_DL\_CS(R5),- :IF BC NO  
 00DO C5 00 00DO 07E7 1411 UCBSB\_DL\_XBA(R5) :SAVE MEMORY EXTENSION BITS  
 00EC C5 00 00EC 07EA 1412 MOVW UCBSW\_DL\_BA(R5),- :  
 07EE 1413 UCBSW\_DL\_SBA(R5) :UPDATE SAVED BUFFER ADDRESS  
 07F1 1414 :  
 51 00D2 C5 96 008C C5 1415 108: CLRB UCBSW\_DA(R5) :UPDATE DESIRED SECTOR TO ZERO  
 52 51 01 06 00000040 8F C1 07F5 1416 ADDL3 #^0100,UCBSW\_DL\_DA(R5),R1 :INCREMENT CYLINDER & SURFACE  
 00BD C5 52 008D 01 06 EF 07FF 1417 EXTZV #6,#1,R1,R2 :EXTRACT DESIRED DISK SURFACE  
 52 51 09 07 00BE C5 52 008E 07 09 EF 0804 1418 MOVB R2,UCBSW\_DA+1(R5) :UPDATE DESIRED DISK SURFACE  
 00CC C5 00 00CC 0809 1419 EXTZV #7,#9,R1,R2 :EXTRACT DESIRED DISK CYLINDER  
 00CO C5 00 00CO 0813 1420 MOVW R2,UCBSW\_DC(R5) :UPDATE DESIRED DISK CYLINDER  
 03 13 FAF8 31 081A 1421 ADDW UCBSW\_DL\_PBCR(R5),- :UPDATE NEG BYTES REMAINING TO XFER  
 081A 1422 UCBSW\_BCR(R5)  
 081C 1423 BEQL RETREG :IF EQL - TRANSFER COMPLETE  
 081C 1424 BRW FDISPATCH :MORE BYTES REMAINING - CONTINUE  
 081F 1425 :  
 081F 1426 : GET STATUS AND RESET ERRORS  
 081F 1427 :  
 081F 1428 :  
 081F 1429 :  
 081F 1430 RETREG: :GET STATUS AND RESET ERRORS  
 081F 1431 :  
 081F 1432 : FOR A READ OPERATION WHEN NO ADAPTER MAPPING IS PRESENT  
 081F 1433 : EMPTY INTERNAL BUFFER INTO USER'S BUFFER FOR LAST READ  
 081F 1434 :  
 015B 30 081F 1435 BSBW DL\_MOVE\_TO\_BUFFER :MOVE LAST READ INTO USER'S BUFFER  
 0822 1436 :  
 0822 1437 SETIPL UCBSB\_FIPL(R5) :MAKE SURE AT FORK IPL (TIMEOUT)  
 04 A4 03 B0 0826 1438 MOVW #RL\_DA#STS!- :PUT GET STATUS IN DAR  
 082A 1439 RL\_DA\_R\_ARL,RL\_DA(R4) :  
 ...

52	08 08 54 52	D4	082A	1440	CLRL	R2	:CLEAR R2 FOR UNIT NUMBER
	64 04 52	F0	082C	1441	INSV	UCBSU_UNIT(R5),#8,#8,R2	:GET UNIT NUMBER
		A9	0832	1442	BISW3	R2,#F-GETSTATUS,RL_CS(R4)	:EXECUTE GET STATUS
00D4 C5	06 A4	30	0836	1443	BSBW	DL_WAIT	:WAIT FOR CONTROLLER
04 A4	0B	B0	0839	1444	MOVW	RL_MP(R4),UCBSU_DL_MP(R5)	:RETRIEVE ERROR REGISTER
		B0	083F	1445	MOVW	#RC_DA_M_RST!-	:PUT GET STATUS & RESET IN DAR
64 04 52	FA19	A9	0843	1446	BISW3	RL_DA_R_STS!RL_DA_M_MRK_RL_DA(R4)	
		30	0847	1448	BSBW	R2,#F-GETSTATUS,RC_TS(R4)	:EXECUTE RESET
			084A	1449		DL_WAIT	:WAIT FOR CONTROLLER
			084A	1450			
			084A	1451			: DETERMINE EXIT - SPECIAL CONDITION, FATAL ERROR, RETRIABLE ERROR, OR SUCCESS
			084A	1452			
			084A	1453			
00D4 C5	05 00	ED	084A	1454	CMPZV	#0,#5,UCBSU_DL_MP(R5),-	:HEADS, BRUSHES, STATE OK?
	1D		0850	1455		#RL_MP_M_BH!RL_MP_M_H0!RL_SLM	
	OE	13	0851	1456	BEQL	1S	:IF EQL - YES, ONLINE
64 A5	0040 8F	AA	0853	1457	BICW	#UCBSM_TIMEOUT,UCBSU_STS(R5)	:CLEAR DEVICE TIME OUT
50	01A4 8F	3C	0859	1458	MOVZWL	#SSS_MEDOFL,R0	:SET MEDIUM OFFLINE STATUS
	FB85	31	085E	1459	BRW	FUNCXT	:RETURN
64 A5	0060 8F	B3	0861	1460	1S:	#UCBSM_POWER!-	:POWER FAIL OR DEVICE TIMEOUT?
			0867	1461	BITW	UCBSM_TIMEOUT,UCBSU_STS(R5)	
			0867	1462	BNEQ	SPECOND	:IF NEQ - YES, SPECIAL CONDITION
6A 00D4 C5	09	E0	0869	1463			
05 00CE C5	0F	E0	086F	1464	BBS	#RL_MP_V_VC,UCBSU_DL_MP(R5),20\$	:IF SET - VOLUME INVALID
	37 00E4 C5	E9	0875	1465	BBS	#RL_CS_V_CE,UCBSU_DL_CS(R5),2\$	:IF SET - RL ERROR
	00000000 GF	16	087A	1466	BLBC	UCBSB_BL_DPP(E(R5),10\$	:IF CLR - NO PURGE ERROR
33 009A C5	0F	E0	0880	1467	2S:	G^ERL5DEVICERR	:ALLOCATE AND FILL ERROR MESSAGE BUFFER
2D 00CE C5	0D	E0	0886	1468	BBS	#IOSV_INHRETRY,UCBSU_FUNC(R5),20\$	:IF SET - RETRY INHIBITED
15 00CE C5	0E	E1	088C	1469	BBS	#RL_CS_V_NXM,UCBSU_DL_CS(R5),20\$	:IF SET - NONEXISTENT MEMORY
06 00D4 C5	0D	E1	0892	1470	BBC	#RL_CS_V_DE,UCBSU_DL_CS(R5),5\$	:IF CLR - NO DRIVE ERRORS
1B 00D4 C5	0A	E0	0898	1471	BBC	#RL_MP_V_WL,UCBSU_DL_MP(R5),4\$	:IF CLR - NOT WRITE LOCKED
00D4 C5	C500 8F	B3	089E	1472	BBS	#RL_MP_V_WGE,UCBSU_DL_MP(R5),20\$	:IF WL & WGE SET - WL ERROR
			08A5	1473	4S:	#RL_MP_M_WDE!-	:WRITE DATA ERROR, OR
			08A5	1474	BITW	RL_MP_M_CHE!-	:CURRENT HEAD ERROR, OR
			08A5	1475		RL_MP_M_WGE!-	:WRITE GATE ERROR, OR
			08A5	1476	BNEQ	RL_MP_M_DSE,UCBSU_DL_MP(R5)	:DRIVE SELECT ERROR?
			08A7	1477		20\$	:IF NEQ - YES
			08A7	1478			
			08A7	1479			
			08A7	1480			: RETRIABLE ERROR EXIT
			08A7	1481			
			08A7	1482			
7E 009C D5	98	08A7	1483	5S:	CVTBL	UCBSL_DPC(R5)-(SP)	:GET BRANCH DISPLACEMENT
009C C5 8E	C0	08AC	1484		ADDL	(SP)+,UCBSL_DPC(R5)	:CALCULATE RETURN ADDRESS - 1
		08B1	1485				
		08B1	1486				
		08B1	1487				: SUCCESSFUL OPERATION EXIT
		08B1	1488				
		08B1	1489				
009C C5	D6	08B1	1490	10\$:	INCL	UCBSL_DPC(R5)	:ADJUST TO CORRECT RETURN ADDRESS
009C D5	17	08B5	1491		JMP	UCBSL_DPC(R5)	:RETURN TO DRIVER
		08B9	1492				
		08B9	1493				
		08B9	1494				: FATAL ERROR EXIT
		08B9	1495				
		08B9	1496				

FADC	31	08B9	1497	208:	BRW	FATALERR		:FATAL ERROR EXIT
		08BC	1498					
		08BC	1499	:				
		08BC	1500	:	SPECIAL CONDITION EXIT (POWER FAILURE OR DEVICE TIMEOUT)			
		08BC	1501	:				
		08BC	1502					
		08BC	1503	SPECOND:				
27 64 A5 05	E0	08BC	1504		BBS	#UCBSV_POWER,UCBSW_STS(R5),PWRFAIL	;IF SET - POWER FAILURE	
00000000'GF	16	08C1	1505				;IF CLR - DEVICE TIMEOUT	
64 A5 0040 BF	AA	08C7	1506		JSB	G\$ERL\$DEVICTMO	;LOG DEVICE TIMEOUT	
50 022C 8F	3C	08CD	1507		BICW	#UCBSM_TIMEOUT,UCBSW_STS(R5)	;CLEAR TIMEOUT STATUS	
0080 C5	97	08D2	1508		MOVZWL	#SSS_TIMEOUT,R0	;SET DEVICE TIMEOUT STATUS	
03	13	08D6	1509		DEC8	UCBSB_ERTCNT(R5)	;ANY ERROR RETRIES REMAINING?	
FASC	31	08D8	1510		BEQL	RESETXFR	;IF EQL - NO	
		08DB	1511		BRW	FDISPATCH	;RETURN	
		08DB	1512					
00C0 53 58 A5	D0	08DB	1513	RESETXFR:				
32 A3	AE	08DF	1514		MOVL	UCBSL_IRP(R5),R3	;RESET TRANSFER BYTE COUNT	
FAFE	31	08E5	1515		MNEGW	IRPSL_BCNTR3,UCBSW_BCR(R5)	;GET ADDRESS OF I/O PACKET	
		08E8	1516		BRW	FUNCXT	;RESET BYTE COUNT	
		08E8	1517				;EXIT	
64 A5 20	AA	08E8	1518	PWRFAIL:				
00D6 C5	BS	08EC	1519		BICW	#UCBSM_POWER,UCBSW_STS(R5)	;POWER FAILURE	
12	13	08F0	1520		TSTW	UCBSW_DL_DPN(R5)	;CLEAR POWER FAILURE BIT	
01	E1	08F2	1521		BEQL	SOS	;ARE UCB RESOURCES ALLOCATED?	
OC 00F6 C5		08F4	1522		BBC	#UCBSV_DL_MAPPING,-	;IF EQL - NO	
		08F8	1523			UCBSW_DL_FLAGS(R5),SOS	;ADAPTER MAPPING?	
		08FE	1524		RELDPR		;IF BC NO	
		0904	1525		RELMPR		;RELEASE DATA PATH	
53 58 A5	D0	090A	1526	508:	RELCHAN		;RELEASE MAP REGISTERS	
2C A3	7D	090E	1527		MOVL	UCBSL_IRP(R5),R3	;RELEASE CHANNEL IF OWNED	
78 A5		0911	1528		MOVG	IRPSL_SVAPTE(R3),-	;GET ADDRESS OF I/O PACKET	
F992	31	0913	1529		BRW	UCBSL_SVAPTE(R5)	;RESTORE TRANSFER PARAMETERS	
			1530			PREPROCESS		;RETURN TO PREPROCESS UCB FIELDS

## .SBTTL INTERRUPT SERVICE ROUTINE

## DLSINT - RL11 INTERRUPT SERVICE ROUTINE

## FUNCTIONAL DESCRIPTION:

THIS ROUTINE IS ENTERED VIA A JSB INSTRUCTION WHEN AN INTERRUPT OCCURS ON AN RL11 DISK CONTROLLER. IF THE INTERRUPT IS NOT EXPECTED, THE UNSOLICITED INTERRUPT ROUTINE DISMISSES THE INTERRUPT. IF THE INTERRUPT IS EXPECTED, DEVICE REGISTERS ARE SAVED AND THE DRIVER IS CALLED AT ITS INTERRUPT RETURN ADDRESS. THE DRIVER FORKS, CAUSING A RETURN TO THIS ROUTINE, WHICH RESTORES GENERAL REGISTERS AND DISMISSES THE INTERRUPT.

## INPUTS:

0916 1548	-	00(SP) - POINTER TO ADDRESS OF THE IDB
0916 1549	-	04(SP) - SAVED R0
0916 1550	-	08(SP) - SAVED R1
0916 1551	-	12(SP) - SAVED R2
0916 1552	-	16(SP) - SAVED R3
0916 1553	-	20(SP) - SAVED R4
0916 1554	-	24(SP) - SAVED R5
0916 1555	-	28(SP) - PC AT THE TIME OF THE INTERRUPT
0916 1556	-	32(SP) - PSL AT THE TIME OF THE INTERRUPT

## OUTPUTS:

DEVICE REGISTERS ARE SAVED, IPL IS LOWERED TO FORK LEVEL, THE INTERRUPT IS DISMISSED. ALL REGISTERS EXCEPT R0-R5 ARE PRESERVED.

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

:-

DLDRIVER  
V04-000

- VAX/VMS RL11/RL01,RL02 DISK DRIVER  
INTERRUPT SERVICE ROUTINE

D 8

16-SEP-1984 00:17:29 VAX/VMS Macro V04-00  
5-SEP-1984 00:12:24 [DRIVER.SRC]DLDRIVER.MAR;1

Page 36  
(1)

DMC  
V04

3F 0959 1589 DL\_UNSOLNT:  
BA 0959 1590 POPR ;UN SOLICITED INTERRUPT  
02 0958 1591 REI ;RESTORE R0-R5  
;RETURN FROM INTERRUPT

095C 1593 .SBTTL REGISTER DUMP ROUTINE

095C 1594 ++

095C 1595 DL\_REGDUMP - REGISTER DUMP ROUTINE

095C 1596

095C 1597 FUNCTIONAL DESCRIPTION:

095C 1598

095C 1599 THIS ROUTINE IS CALLED TO SAVE THE DEVICE REGISTERS AND UBA RESOURCE  
095C 1600 REGISTERS IN A SPECIFIED BUFFER. IT IS CALLED FROM THE DEVICE ERROR  
095C 1601 LOGGING ROUTINE AND FROM THE DIAGNOSTIC BUFFER FILL ROUTINE.

095C 1602

095C 1603 INPUTS:

095C 1604

095C 1605

095C 1606

095C 1607

095C 1608

095C 1609

095C 1610

095C 1611

095C 1612

095C 1613

095C 1614

095C 1615

095C 1616

095C 1617

095C 1618

DL\_REGDUMP:  
 1619 MOVL #<RL\_NUM\_REGS+5>, (R0)+ ;REGISTER DUMP ROUTINE  
 1620 MOVAL UCB\$W DL-CS(R5), R1 ;INSERT NUMBER OF REGISTERS  
 1621 MOVZBL #RL\_NUM\_REGS, R2 ;GET ADDRESS OF SAVED DEVICE REGISTERS  
 1622 10\$: MOVZWL (R1)+, (R0)+ ;GET NUMBER OF DEVICE REGISTERS TO MOVE  
 1623 SOBGTR R2, 10\$ ;DUMP REGISTER IN BUFFER  
 1624 MOVZWL (R1)+, (R0)+ ;IF GTR - STILL MORE TO MOVE  
 1625 MOVL (R1)+, (R0)+ ;DUMP DATAPATH NUMBER  
 1626 MOVL (R1)+, (R0)+ ;DUMP DATAPATH REGISTER  
 1627 MOVL (R1)+, (R0)+ ;DUMP FINAL MAP REGISTER  
 1628 MOVZBL (R1)+, (R0)+ ;DUMP PREVIOUS MAP REGISTER  
 1629 RSB ;DUMP DATAPATH PURGE ERROR REGISTER  
 1630

51	80	09	00	095C	1619
	00CE	C5	DE	095F	1620
	52	04	9A	0964	1621
	80	81	3C	0967	1622
	FA	52	F5	096A	1623
	80	81	3C	096D	1624
	80	81	DO	0970	1625
	80	81	DO	0973	1626
	80	81	DO	0976	1627
	80	81	9A	0979	1628
			05	097C	1629
				097D	1630

097D 1632 .SBTTL MOVE TO USER BUFFER ROUTINE  
 097D 1633 :++  
 097D 1634 DL\_MOVE\_TO\_BUFFER - MOVE TO USER BUFFER  
 097D 1635 FUNCTIONAL DESCRIPTION:  
 097D 1636 THIS ROUTINE MOVES DATA BETWEEN THE PHYSICALLY CONTIGUOUS BUFFER AND  
 097D 1637 THE USER'S BUFFER.  
 097D 1638 INPUTS:  
 097D 1639 R5 - UCB ADDRESS  
 097D 1640 OUTPUTS:  
 097D 1641 097D 1642 097D 1643 097D 1644 097D 1645 097D 1646 097D 1647 097D 1648 097D 1649 097D 1650 097D 1651 097D 1652 097D 1653 DATA MOVE BETWEEN THE PHYSICALLY CONTIGUOUS BUFFER AND THE USER'S BUFFER.  
 097D 1654 REGISTER'S R0,R1, AND R2 ARE DESTROYED  
 097D 1655 :--  
 01 E0 097D 1656 DL\_MOVE\_TO\_BUFFER:  
 0093 C5 0C 91 0983 1656 BBS #UCBSV\_DL\_MAPPING,-  
 00C9 C5 00 E0 098A 1657 CMPB #CDF\_READDATA,UCBSB\_CEX(R5);READ DATA OPERATION?  
 00E0 C5 27 098F 1658 BNEQ 10\$  
 00E0 C5 D5 0990 1660 BBS #0\_UCBSB\_DL\_DCHEK(R5),-  
 50 00DC C5 22 13 0994 1661 TSTL UCBSA\_DL\_MOVRTN(R5)  
 51 00EE C5 D0 0996 1662 BEQL 20\$  
 52 00CC C5 3C 09A0 1663 MOVL UCBSL\_DL\_BUFADR(R5),R0  
 00E0 D5 16 09A5 1664 MOVL UCBSA\_DL\_BUF\_VA(R5),R1  
 00DC C5 50 D0 09A9 1665 MOVZWL UCBSW\_DL\_PBCR(R5),R2  
 00000000'GF 9E 09AE 1666 JSB UCBSA\_DL\_MOVRTN(R5)  
 00E0 C5 09B4 1668 MOVL R0\_UCBSL\_DL\_BUFADR(R5)  
 05 09B7 1669 10\$: MOVAB G^IOCSMOVTOUSER,-  
 00000000'GF 9E 09B8 1670 RSB UCBSA\_DL\_MOVRTN(R5)  
 00E0 C5 09BE 1671 20\$: MOVAB G^IOCSMOVTOUSER,-  
 05 09C1 1672 RSB UCBSA\_DL\_MOVRTN(R5)  
 09C2 1674 RSB

:BUFFER MOVE ROUTINE  
 :ADAPTER MAPPING?  
 :IF BS YES NOTHING TO MOVE  
 :IF CDF READ DATA OPERATION?  
 :IF NEQ NOT A READ  
 :DATA CHECK IN PROGRESS?  
 :IF BS YES NOTHING TO MOVE  
 :ANYTHING TO MOVE?  
 :IF EQL NO  
 :GET USER BUFFER POINTER  
 :GET PHYSICALLY CONTIGUOUS BUFFER ADDRESS  
 :GET NUMBER OF BYTES TO TRANSFER  
 :CALL MOVE ROUTINE  
 :SAVE INTERNAL BUFFER POINTER  
 :SET NEXT MOVE ROUTINE TO BE USED  
 :RETURN  
 :SET NEXT MOVE ROUTINE TO BE USED  
 :RETURN

	09C2	1676		.SBTTL MOVE FROM USER BUFFER ROUTINE
	09C2	1677	++	
	09C2	1678		DL_MOVE_FROM_BUFFER - MOVE FROM USER BUFFER
	09C2	1679		
	09C2	1680		FUNCTIONAL DESCRIPTION:
	09C2	1681		THIS ROUTINE MOVES DATA BETWEEN THE PHYSICALLY CONTIGUOUS BUFFER AND
	09C2	1682		THE USER'S BUFFER.
	09C2	1683		
	09C2	1684		INPUTS:
	09C2	1685		
	09C2	1686		RS - UCB ADDRESS
	09C2	1687		
	09C2	1688		OUTPUTS:
	09C2	1689		
	09C2	1690		DATA MOVE BETWEEN THE PHYSICALLY CONTIGUOUS BUFFER AND THE USER'S BUFFER.
	09C2	1691		REGISTER'S R0,R1, AND R2 ARE DESTROYED
	09C2	1692		
	09C2	1693		
	09C2	1694		
	09C2	1695	--	
	09C2	1696		
	09C2	1697	DL_MOVE_FROM_BUFFER:	
2E 00F6	01	E0	09C2	BBS #UCBSV_DL_MAPPING,-
0093 CS	C5	09C4	1698	UCBSW_DL_FLAGS(R5),10\$
08	91	09C8	1699	:ADAPTER MAPPING?
27	12	09CD	1700	CMPB #CDF_WRITEDATA,UCB\$B_CEX(R5)
00C9 CS	00	E0	09CF	BNEQ 10\$
21		09D4	1701	:IF BS YES NOTHING TO MOVE
50 00DC CS	D0	09D5	1702	BBS #0_UCBSB_DL_DCHEK(R5),-
51 00EE CS	D0	09DA	1703	10\$
52 00CC CS	3C	09DF	1704	:DATA CHECK IN PROGRESS?
00E0 DS	16	09E4	1705	MOVL UCBSL_DL_BUFADR(R5),R0
00DC CS 50	D0	09E8	1706	MOVL UCBSA_DL_BUF_VA(R5),R1
00000000'GF	9E	09ED	1707	MOVZWL UCBSW_DL_PBCR(R5),R2
00E0 CS	09F3	1708		:GET USER BUFFER POINTER
	09F6	1709	JSB UCBSA_DL_MOVRTN(R5)	:GET PHYSICALLY CONTIGUOUS BUFFER ADDRESS
	09F7	1710	MOVL R0,UCBSL_DL_BUFADR(R5)	:GET NUMBER OF BYTES TO TRANSFER
	09F7	1711	MOVAB G^IOCSMOVFRUSER2,-	:CALL MOVE ROUTINE
	09F7	1712	UCBSA_DL_MOVRTN(R5)	:SAVE INTERNAL BUFFER POINTER
	09F7	1713	RSB	:SET NEXT MOVE ROUTINE TO BE USED
	09F7	1714	DL_END: .END	:RETURN
				:ADDRESS OF LAST LOCATION IN DRIVER

\$SS	= 00000020	R	02	DL_INT	00000916	RG	03
\$S\$BASE	= 00000001			DL_MOVE_FROM_BUFFER	000009C2	R	03
\$S\$DISPL	= 0000000A			DL_MOVE_TO_BUFFER	0000097D	R	03
\$S\$GENSW	= 00000001			DL_REGDUMP	0000095C	R	03
\$S\$HIGH	= 00000009			DL_RLOX_INIT	00000143	R	03
\$S\$LIMIT	= 00000008			DL_RL11_INIT	000000FC	R	03
\$S\$LOW	= 00000001			DL_STARTIO	000002A8	R	03
\$S\$MNSW	= 00000001			DL_UNSLNT	00000959	R	03
\$S\$MXSM	= 00000001			DL_WAIT	00000263	RR	03
\$S\$OP	= 00000002			DO_FUNCTION	0000035C	R	03
ACPSACCESS	*****	X	03	DPTSC_LENGTH	= 00000038		
ACPSDEACCESS	*****	X	03	DPTSC_VERSION	= 00000004		
ACPSMODIFY	*****	X	03	DPT\$INITAB	= 00000038	R	02
ACPSMOUNT	*****	X	03	DPTSM_SVP	= 00000002		
ACPSREADBLK	*****	X	03	DPTSREINITAB	= 00000073	R	02
ACPSWRITEBLK	*****	X	03	DPTSTAB	00000000	RR	02
AT8_UBA				DRCLR	00000477	RR	03
AVAILABLE	= 00000001			DRVCLR	0000035C	R	03
BUGS UNSUPRTCPU	*****	X	03	DTS_RL01	= 00000009		
CDF_AVAILABLE	= 00000011			DTS_RL02	= 0000000A		
CDF_DRVCLR	= 00000004			DYNSC_CRB	= 00000005		
CDF_NOP	= 00000010			DYNSC_DDB	= 00000006		
CDF_OFFSET	= 00000006			DYNSC_DPT	= 0000001E		
CDF_PACKACK	= 00000008			DYNSC_UCB	= 00000010		
CDF_READDATA	= 0000000C			EMBSL_DV_REGS	= 0000004E		
CDF_READHEAD	= 0000000E			ERLSDEVICERR	*****	X	03
CDF_RECAL	= 00000003			ERLSDEVICTMO	*****	X	03
CDF_RELEASE	= 00000005			EXESABORTIO	*****	X	03
CDF_RETCENTER	= 00000007			EXESALOPHYCNTG	*****	X	03
CDF_SEARCH	= 00000009			EXESGB_CPUTYPE	*****	X	03
CDF_SEEK	= 00000002			EXESGL_TENUSEC	*****	X	03
CDF_UNLOAD	= 00000001			EXESGL_UBDELAY	*****	X	03
CDF_WRITECHECK	= 0000000A			EXESIOPORK	*****	X	03
CDF_WITITEDATA	= 0000000B			EXESLCLDISKVALID	*****	X	03
CDF_WRITEHEAD	= 0000000D			EXESONEPARM	*****	X	03
CRBSL_B_MASK	= 0000000E			EXESPWTIMCHK	*****	X	03
CRBSL_AUXSTRUC	= 00000010			EXESSENSEMODE	*****	X	03
CRBSL_INTD	= 00000024			EXESSETCHAR	*****	X	03
CRBSV_BSY	= 00000000			EXESZEROPARM	*****	X	03
DATACRECK	= 000007C5	R	03	FATALERR	00000398	R	03
DC8_DISK	= 00000001			FDISPATCH	00000317	R	03
DDBSL_CART	= 00000002			FEKL	00000439	RR	03
DDBSL_ACPD	= 00000010			FTAB	00000038	R	03
DDBSL_DDT	= 0000000C			FUNCTAB_LEN	= 000000A0		
DEVSM_AVL	= 00040000			FUNCXT	= 000003E6	R	03
DEVSM_DIR	= 00000008			F_AVAILABLE	= 00000000		
DEVSM_ELG	= 00400000			F_DRVCLR	= 00000004		
DEVSM_FOD	= 00004000			F_GETSTATUS	= 00000004		
DEVSM_IDV	= 04000000			F_NOP	= 00000000		
DEVSM_NNM	= 00000200			F_OFFSET	= 00000000		
DEVSM_ODV	= 08000000			F_PACKACK	= 00000004		
DEVSM_RND	= 10000000			F_READDATA	= 0000000C		
DEVSM_SHR	= 00010000			F_READHEAD	= 00000008		
DL\$DDT	= 00000000	RG	03	F_RECAL	= 00000000		
DL_ALIGN	= 00000298	R	03	F_RELEASE	= 00000000		
DL_END	= 000009F7	R	03	F_RETCENTER	= 00000000		
DL_FUNCTABLE	= 0000005C	R	03				

F_SEARCH	= 00000000		IRPSS_FCODE	= 00000006
F_SEEK	= 00000006		IRPSV_DIAGBUF	= 00000007
F_UNLOAD	= 00000000		IRPSV_FCODE	= 00000000
F_WRITECHECK	= 00000002		IRPSV_PHYSIO	= 00000008
F_WRITEDATA	= 0000000A		IRPSW_BCNT	= 00000032
F_WRITEHEAD	= 00000000		IRPSW_FUNC	= 00000020
IDBSL_CSR	= 00000000		IRPSW_STS	= 0000002A
IDBSL_OWNER	= 00000004		MASKH	= 00000008
IMMED	0000047B	R 03	MASKL	= 04000000
IOSM_DATACHECK	= 00004000		MMGSGL_SPTBASE	***** X 03
IOSV_DATACHECK	= 0000000E		NOP	0000035C R 03
IOSV_INHRETRY	= 0000000F		NORMAL	00000389 R 03
IOS_ACCESS	= 00000032		PACKACK	00000362 R 03
IOS_ACPCONTROL	= 00000038		POSIT	000004AE R 03
IOS_AVAILABLE	= 00000011		PRS_IPL	= 00000012
IOS_CREATE	= 00000033		PRS_SID_TYP730	= 00000003
IOS_DEACCESS	= 00000034		PRS_SID_TYP750	= 00000002
IOS_DELETE	= 00000035		PRS_SID_TYP780	= 00000001
IOS_DRVCLR	= 00000004		PRS_SID_TYP785	= 00000009
IOS MODIFY	= 00000036		PRS_SID_TYP790	= 00000004
IOS_MOUNT	= 00000039		PRS_SID_TYPMAX	= 00000008
IOS_NOP	= 00000000		PRS_SID_TYPUV1	= 00000007
IOS_PACKACK	= 00000008		PREPROCESS	000002A8 R 03
IOS_READHEAD	= 0000000E		PTESS_PFN	= 00000015
IOS_READLBLK	= 00000021		PWRFAIL	000008E8 R 03
IOS_READPBLK	= 0000000C		READDATA	00000379 R 03
IOS_READVBLK	= 00000031		READHEAD	00000372 R 03
IOS_SEEK	= 00000002		RESETXFR	000008DB R 03
IOS_SENSECHAR	= 00000018		RETREG	0000081F R 03
IOS_SENSEMODE	= 00000027		RETRYERR	0000038F R 03
IOS_SETCHAR	= 0000001A		RL_BA	= 00000002
IOS_SETMODE	= 00000023		RL_BAE	= 00000008
IOS_UNLOAD	= 00000001		RL_CS	= 00000000
IOS_VIRTUAL	= 0000003F		RL_CS_M_CE	= 00008000
IOS_WRITECHECK	= 0000000A		RL_CS_M_CRDY	= 00000080
IOS_WRITELBLK	= 00000020		RL_CS_M_DRDY	= 00000001
IOS_WRITEPBLK	= 0000000B		RL_CS_M_IE	= 00000040
IOS_WRITEVBLK	= 00000030		RL_CS_V_CE	= 0000000F
IOC\$DIAGBUFILL	*****	X 03	RL_CS_V_CRC	= 0000000B
IOC\$LOADUBAMAP	*****	X 03	RL_CS_V_DE	= 0000000E
IOC\$MNTPVER	*****	X 03	RL_CS_V_NXM	= 0000000D
IOC\$MOVFRUSER	*****	X 03	RL_CS_V_OPI	= 0000000A
IOC\$MOVFRUSER2	*****	X 03	RL_DA	= 00000004
IOC\$MOVTOUSER	*****	X 03	RL_DA_M_MRK	= 00000001
IOC\$MOVTOUSER2	*****	X 03	RL_DA_M_RST	= 00000008
IOC\$PURGDATA	*****	X 03	RL_DA_M_STS	= 00000002
IOC\$RELCHAN	*****	X 03	RL_MP	= 00000006
IOC\$RELDAT	*****	X 03	RL_MP_M_BH	= 00000008
IOC\$RELMAPREG	*****	X 03	RL_MP_M_CHE	= 00004000
IOC\$REQCOM	*****	X 03	RL_MP_M_DSE	= 0000100
IOC\$REQDAT	*****	X 03	RL_MP_M_HO	= 00000010
IOC\$REQMAPREG	*****	X 03	RL_MP_M_TYP	= 00000080
IOC\$REQPCCHAN	*****	X 03	RL_MP_M_WDE	= 00008000
IOC\$RETURN	*****	X 03	RL_MP_M_WGE	= 0000400
IOC\$WFICKPCH	*****	X 03	RL_MP_V_VC	= 00000009
IRPSL_MEDIA	= 00000038		RL_MP_V_WGE	= 0000000A
IRPSL_SVAPTE	= 0000002C		RL_MP_V_WL	= 0000000D

DLDRIER  
Symbol table

- VAX/VMS RL11/RL01,RL02 DISK DRIVER

J 8

16-SEP-1984 00:17:29 VAX/VMS Macro V04-00  
5-SEP-1984 00:12:24 [DRIVER.SRC]DLDRIER.MAR;1Page 42  
(1)

RL_NUM_REGS	= 00000004		UCBSV_DL_MAPPING	= 00000001
RL_SLM	= 00000005		UCBSV_INT	= 00000001
SEEK	= 0000035C R 03		UCBSV_POWER	= 00000005
SIZ...	= 00000001		UCBSV_VALID	= 00000008
SPECOND	= 000008BC R 03		UCBSW_BCNT	= 0000007E
SSS_CTRLERR	= 00000054		UCBSW_BCR	= 000000C0
SSS_DATACHECK	= 0000005C		UCBSW_BOFF	= 0000007C
SSS_DRVERR	= 0000008C		UCBSW_CYLINDERS	= 00000046
SSS_IVBUFLN	= 0000034C		UCBSW_DA	= 000000BC
SSS_MEDOFL	= 000001A4		UCBSW_DC	= 000000BE
SSS_NORMAL	= 00000001		UCBSW_DEVBUFSIZ	= 00000042
SSS_PARITY	= 000001F4		UCBSW_DEVSTS	= 00000068
SSS_TIMEOUT	= 0000022C		UCBSW_DL_BA	= 000000D0
SSS_VOLINV	= 00000254		UCBSW_DL_CS	= 000000CE
SSS_WRLCK	= 0000025C		UCBSW_DL_DA	= 000000D2
UCBSA_DL_BUF_PA	000000F2		UCBSW_DL_DB	= 000000E5
UCBSA_DL_BUF_VA	000000EE		UCBSW_DL_DPN	= 000000D6
UCBSA_DL_MOVRTN	000000E0		UCBSW_DL_FLAGS	= 000000F6
UCBSB_CEX	= 00000093		UCBSW_DL_MP	= 000000D4
UCBSB_DEVCLASS	= 00000040		UCBSW_DL_PBCR	= 000000CC
UCBSB_DEVTYPE	= 00000041		UCBSW_DL_SBA	= 000000EC
UCBSB_DIPL	= 0000005E		UCBSW_FUNC	= 0000009A
UCBSB_DL_DCHEK	= 000000C9		UCBSW_OFFSET	= 000000C8
UCBSB_DL_DPPE	000000E4		UCBSW_STS	= 00000064
UCBSB_DL_XBA	000000EB		UCBSW_UNIT	= 00000054
UCBSB_ERTCNT	= 00000080		UNLOAD	0000036A R 03
UCBSB_ERTMAX	= 00000081		UPDATE	000007DB R 03
UCBSB_FEX	= 00000092		VASM_BYTE	= 000001FF
UCBSB_FIPL	= 00000008		VASS_VPN	= 00000015
UCBSB_SECTORS	= 00000044		VASV_VPN	= 00000009
UCBSB_TRACKS	= 00000045		VECSB_DATAPATH	= 00000013
UCBSK_DL_BUFSZ	= 00000014		VECSL_IDB	= 00000008
UCBSK_DL_LEN	000000F8		VECSL_INITIAL	= 0000000C
UCBSK_LCL_DISK_LENGTH	= 000000CC		VECSL_UNITINIT	= 00000018
UCBSL_CRB	= 00000024		VECSS_DATAPATH	= 00000005
UCBSL_DEVCHAR	= 00000038		VECSS_MAPREG	= 0000000F
UCBSL_DEVCHAR2	= 0000003C		VECSV_DATAPATH	= 00000000
UCBSL_DL_BUFADR	000000DC		VECSV_MAPREG	= 00000000
UCBSL_DL_DPR	000000D8		VECSW_MAPREG	= 00000010
UCBSL_DL_FMPR	000000DC		WRITECHECK	00000372 R 03
UCBSL_DL_PMPR	000000E0		WRITEDATA	00000379 R 03
UCBSL_DL_SVAPTE	000000D8		XFER	0000061A R 03
UCBSL_DPC	= 0000009C			
UCBSL_FPC	= 0000000C			
UCBSL_FR3	= 00000010			
UCBSL_IRP	= 00000058			
UCBSL_MAXBLOCK	= 000000B0			
UCBSL_MEDIA	= 000000BC			
UCBSL_MEDIA_ID	= 0000008C			
UCBSL_SVAPTE	= 00000078			
UCBSM_DIAGBUF	= 00000002			
UCBSM_NOCNVRT	= 00000004			
UCBSM_ONLINE	= 00000010			
UCBSM_POWER	= 00000020			
UCBSM_TIMEOUT	= 00000040			
UCBSM_VALID	= 00000800			
UCBSV_DL_22BIT	= 00000000			

```
+-----+
! Psect synopsis !
+-----+
```

PSECT name	Allocation	PSECT No.	Attributes											
ABS .	00000000	00 ( 0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE	
\$ABSS	000000FA	01 ( 1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE	
\$\$\$105_PROLOGUE	00000088	02 ( 2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE	
\$\$\$115_DRIVER	000009F7	03 ( 3.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	LONG	

```
+-----+
! Performance indicators !
+-----+
```

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.05	00:00:00.55
Command processing	128	00:00:00.39	00:00:01.95
Pass 1	642	00:00:20.98	00:01:13.05
Symbol table sort	0	00:00:02.56	00:00:09.30
Pass 2	308	00:00:04.50	00:00:21.00
Symbol table output	38	00:00:00.23	00:00:01.81
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1150	00:00:28.74	00:01:47.68

The working set limit was 2250 pages.

171426 bytes (335 pages) of virtual memory were used to buffer the intermediate code.

There were 130 pages of symbol table space allocated to hold 2340 non-local and 91 local symbols.

1714 source lines were read in Pass 1, producing 24 object records in Pass 2.

67 pages of virtual memory were used to define 62 macros.

```
+-----+
! Macro library statistics !
+-----+
```

Macro library name	Macros defined
\$255SDUA28:[SYS.OBJ]LIB.MLB;1	40
\$255SDUA28:[SYSLIB]STARLET.MLB;2	11
TOTALS (all libraries)	51

2659 GETS were required to define 51 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:DLDRIER/OBJ=OBJ\$:DLDRIER MSRC\$:DLDRIER/UPDATE=(ENHS:DLDRIER)+EXECMLS/LIB

0109 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

